Adaptive Support for Human-Computer Teams

van Maanen, P.

2010

document version
Publisher's PDF, also known as Version of record

Adaptive Support for Human-Computer Teams

Exploring the Use of Cognitive Models of Trust and Attention

Peter-Paul van Maanen

SIKS Dissertation Series No. 2010-52

The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Graduate School for Information and Knowledge Systems.

Promotiecommissie:

dr. J.M. Bradshaw	(Florida Institute for Human and Machine Cognition)
dr. R. Falcone	(Institute of Cognitive Sciences and Technologies)
prof.dr. J.-J.Ch. Meyer	(Utrecht University, Turing Institute Almere, TNO Human Factors)
prof.dr. M.A. Neerincx	(TNO Human Factors, Delft University of Technology)
dr. M.C. Schut	(Vrije Universiteit Amsterdam)

ISBN 978-90-865-9516-7

Cover by Marco van Vugt and published by Ipskamp Drukkers B.V., Enschede.

VRIJE UNIVERSITEIT

Adaptive Support for Human-Computer Teams

Exploring the Use of Cognitive Models of Trust and Attention

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad Doctor aan
de Vrije Universiteit Amsterdam,
op gezag van de rector magnificus
prof.dr. L.M. Bouter,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
van de faculteit der Exacte Wetenschappen
op donderdag 9 december 2010 om 9.45 uur
in de aula van de universiteit,
De Boelelaan 1105

door

Peter-Paul van Maanen

geboren te Raamsdonk

promotor: prof.dr. J. Treur
copromotor: dr. T. Bosse

Contents

Part I

Introduction and Methodology

Chapter 1

Introduction

1 Problem Statement

In highly automated domains such as aviation, air traffic control, nuclear power plants and the military, there is an alarming amount of challenges: More complex missions, less manning, higher information density, increased computer autonomy, more ambiguity, more time pressure and higher cooperation demands are causing a very spacious gap between the automated (computers) and the non-automated (humans). Simply put, one of the main problems is that the non-automated as well as the automated are insufficiently aware of each other's capabilities and limitations, while they heavily depend on one another.

Cooperating humans behave socially, i.e., they estimate the other's need for assistance and adapt their support to this estimation. Though state-of-the-art human-computer interfaces more and more contain user models to pro-actively adapt their support, unfortunately in many cases human-like social behavior is nearly absent or still underdeveloped. Especially interfaces that comprise dynamic and real-time computer support adaptation to the current state of humans (as opposed to using predefined user profiles) can be seen as relatively new.

In critical situations, non-existence of social behavior in the interaction between humans and computers can have devastating effects: A famous example is the aircraft pilot being assisted by several support systems in his cockpit at the same time. This abundance of automated support often leads to information overload, possibly resulting in overlooking important information about the failure of a plane engine. Another famous example is the same aircraft pilot over-relying on the auto-pilot while the aircraft's support systems do not take into account the possibility of this complacency. The problem brought forward in these two examples is that pilots, or humans in general, and their support systems are insufficiently aware of the dangers as a result of each other's limitations. Taking the possibility of these limitations into account, could result in much better human-computer cooperative behavior.

This dissertation is about overcoming the above stated problem by increasing the reasoning capabilities of support systems with respect to their own limitations and especially those of their human users (the human factor). System awareness of, and adaptation to, limitations in human-computer cooperation can lead to more social and therefore cooperative behavior of the supporting system. Support systems could for example be aware of information overload, over- and under-trust, complacency, confirmation and automation biases and cognitive under- or over-load. Currently, humans need to specify when support systems are needed, considering the type of support they provide. But in the near future, socially capable support systems will also be able to determine when and in what way they should be used. They adapt pro-actively to the situation and user at hand. Especially in time constrained situations, this would relieve the user of the difficult task to configure support systems appropriately, given the current situation. This could lead to better performances due to freed cognitive resources or due to the system's understanding of limitations the user would otherwise be unaware of.

2 Research Objective

The objective of the research reported in this thesis is to investigate means for integrating knowledge of the human factor in human-computer cooperation into the reasoning capabilities of support systems. This is done to reduce the amount of problems caused by insufficient mutual understanding of the capabilities and limitations of humans and of support systems. The studies described in this thesis are mainly concerned with support systems that are part of highly automated environments as is described in Section 1. The increased reasoning capabilities of support systems are reached by incorporating executable cognitive models, which describe human cognition as accurately as possible, including its limitations, into these systems. Subsequently, these executable cognitive models are used to detect occurrences of limitations. Such detections are then used as triggers for adaptation of the support to the human need for assistance, ideally resulting in an increase, or prevention of a decrease, of human-computer team performance. The specific adaptive support explored in this thesis focuses on *adaptive autonomy* and *decision support*. The specific cognitive models explored in this thesis focus on *trust* and *attention*. These types of adaptive support and cognitive models, and the used support system design, are further explained in Section 3.

3 Background

3.1 Adaptive Support for Human-Computer Teams

One way of adapting to the human need for assistance is to design systems that let automation determine the division of work between humans and computers. Given a set of tasks to be executed, such work division is defined as the allocation of either the human or computer to specific subsets of this set of tasks. The allocation of tasks by the computer can be seen as *adaptive autonomy*: high computer autonomy is equivalent to large portions of these tasks allocated to the computer, and low computer autonomy equivalent to small portions. There are generally speaking two reasons for the adaptation of computer autonomy.

The first reason is that the situation at hand can be subject to change: the appropriate work division is dependent on which subtasks currently have priority to be executed given the state of the (outside) world. For instance, in the case of a classification task, when it is important to classify a certain object and this object has not been classified yet, the adaptive system could indicate this by highlighting this specific object. In this way, in fact, the support system is advising the human to do something about it, i.e., it is trying to allocate a subtask to the human (low computer autonomy). The support system could also have taken care of it and thereby allocating the subtask to itself (high computer autonomy). One could say that each support system is a task allocation mechanism, but mostly by only allocating tasks to humans. Support systems are often not intelligent enough to do a (complex) task themselves, but *are* able to provide relevant information or advice, leaving the final decision to humans (which in fact is *decision support*).

The second reason for computer adaptation is that appropriate task allocations depend

on the state and the capabilities of the involved agents (human or computer) to which those tasks are to be allocated. For instance, a human may be overloaded with work such that currently it is best to allocate tasks to a computer. Task re-allocations for this reason, need to have some sense of how human cognition works in order to determine states like 'cognitively overloaded' or 'inattentive to a certain object'. This 'sense' can be accomplished by the integration of models of human cognition, describing concepts such as 'cognitive overload', into support systems.

3.2 Exploring the Use of Cognitive Models of Trust and Attention

The cognitive models explored in this thesis for the above explained adaptive support focus on *trust* and *attention*. There are many more cognitive functions, concepts or processes that would be very good candidates for the purpose of adapting automated support to the human state and capabilities, but it was simply chosen to only focus on these two since they are very important in many tasks involved with human-computer interaction.

Trust is one of the primary regulators for taking information into account while reasoning, making decisions or generating plans. It also regulates whether a human decides to rely on another (human or computer) or accepts, for instance, an increased level of autonomy of the support system. Being able to describe trust accurately could revolutionize adaptive support systems, because knowledge of acceptance, reliance or impact of information on decisions and plans could then be used to determine when and how to communicate information or when and how to change autonomy. This can be done in such a way that the support is optimally accepted, used and appropriately relied upon. For instance, new support systems would advise against relying on automatic pilots when they estimate this is inappropriate given the current weather conditions. Another possibility is to communicate information more intrusively when it is expected an operator distrusts the system but at the same time both the reliability and urgency of the information is estimated to be very high. The system could for instance give more arguments to convince the operator of this reliability and urgency.

Attention is the one cognitive process or state which is involved in the selection and understanding of information from the 'external world' (overt attention) and the 'internal world' (covert attention). This means that attention is broader than just where somebody is looking at; it also determines of which concepts or subjects one is aware. If support systems are able to identify the concepts or subjects a human is attending to, it could also revolutionize the effectiveness of the support: it would not be doing things which are not relevant anymore. The given support would be fine-tuned to the concepts or subjects the human user is attending to. For instance, when many contacts on a radar screen need to be monitored, the detection of to which contacts a radar operator is attending can give an indication to which other contacts the support system could attend, with significantly less required interventions by the operator.

The approach used in the development of such adaptive systems is based on a comparison between the estimation of the cognitive state of the human and some normative cognitive state. If there exists a large enough discrepancy between the two, this can lead to an adaptation of the given support. For instance, in the above example of the radar operator monitoring contacts, the estimated attentional state of the operator could indicate

that the operator is not paying attention to a certain contact, but the normative cognitive state is indicating that he should. This would result in a detected discrepancy and some intervention can be triggered. For instance, an intervention might be an increase of autonomy: the computer takes over part of the operator's task to deal with suspicious contacts. This obviously raises ethical questions related to whether humans are still responsible for actions performed by autonomous systems. For discrepancies between estimated and normative trust, an adaptation could be for instance to ask humans whether they think it is sensible not to take certain information into account.

4 Support System Design

The number of possible applications of the described approach seems infinite and it is impossible to deal with many of them in one thesis. It has been applied five times: Trust model-based support systems have been studied using a pattern learning task and a classification task. Attention model-based support systems have been studied using an air traffic control task, naval tactical picture compilation task and a shooting game task. The focus in these task environments has been primarily on the increase or decrease of computer autonomy (taking over or delegating (sub)tasks) and the informing of discrepancies between estimated and normative cognitive states (manipulation of trust and attention through advice). For each of these studies the same support system design has been used. This support system design is shown in Figure 1 and is further described below.

(1) **Human (team):** Each support system assists either one or multiple humans in a team that have to perform a certain computer task.

(2) **Human-computer interface:** The human is able to interact with the computer through a human-computer interface. This interface is build on a normal personal computer with one computer screen, or more computer screens in the case of having two or more tasks at the same time. The human can *inform* the interface with his wishes related to the task he is performing. Vice versa, the interface *informs* the human of any important information.

(3) **Task environment:** The task the human has to perform is a task in which either trust or attention plays a key role. For all of these tasks it holds that there is limited time available and mostly the human is responsible for multiple subtasks at the same time. The type of tasks used for the research described in this thesis do not require any direct interventions by a human in the task environment and are always executed via the computer interface.

(4) **Task support component:** The task support component assists the human in his task and can *change* the task environment (or world state) directly based on its own reasoning the observed *environmental data*. The task support component can *inform* and *be informed* through the human-computer interface, i.e., as a way to for instance give advice to the human or to get orders from the human, respectively, with respect to the task.

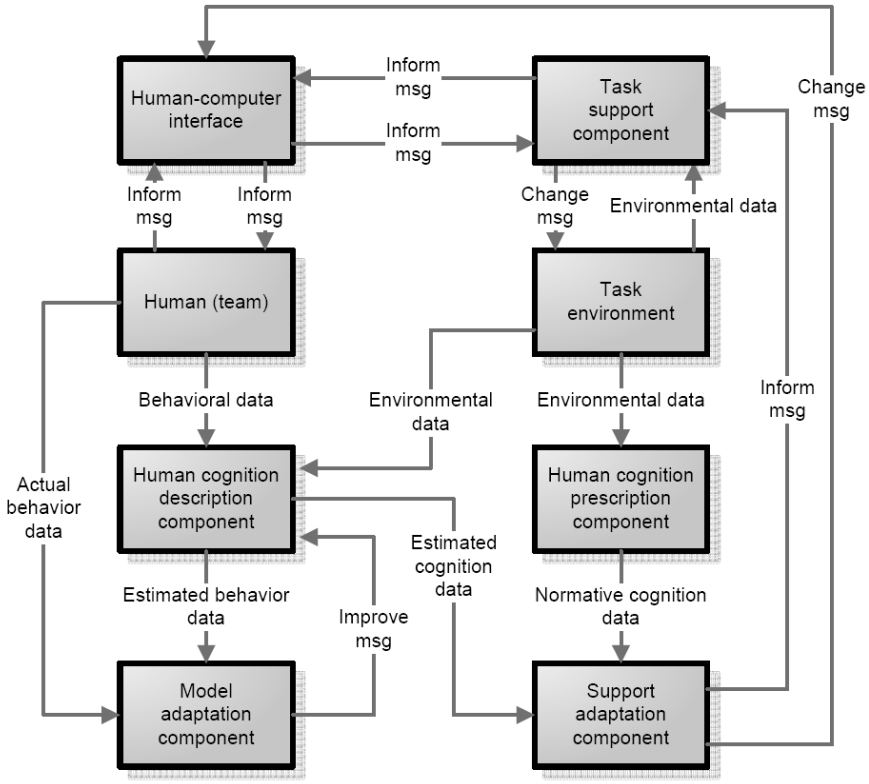


Figure 1: The support system design used in this thesis.

So far, the different components of the support system design are very similar to most other support systems. There are no models that monitor or estimate human cognition in order to use this estimation for triggering adaptation to the human state. This is done through the remainder of the components in Figure 1, explained below.

(5) **Human cognition description component:** First of all there is the human cognition description component which generates its output using a *descriptive cognitive model*. This model uses *behavioral data* combined with *environmental data* to generate *estimated behavior data* and *estimated cognition data*. The estimated behavior data represent the model's expectations that certain behavior (s.a. an action or a sequence of actions) just has taken place (a description of the current situation) or is going to take place in the very near future (a prediction of the next situation). The estimated cognition data represent certain covert cognitive states of the current or next situation. Actual human behavior can be measured, but actual human cognition not.

(6) **Human cognition prescription component:** The human cognition prescription component generates its output using a *prescriptive cognitive model*. This model generates *normative cognition data* using solely *environmental data*, whereas the descriptive cognitive model requires behavioral data as well to determine its estimation. The normative cognition data is meant as a prescribed or suggested cognitive state for the human in order to perform the task optimally.

(7) **Model adaptation component:** The model adaptation component compares the estimated behavior data from the descriptive cognitive model with the *actual behavior data* measured from the human. These measures can be both subjective, such as those based on questionnaires, as well as objective, such as those based on data from sensors (e.g., an eye-tracker or mouse). The model adaptation component can use this comparison to *improve* the descriptive cognitive model in (semi)real-time: given actual and estimated data one can learn the proper parameters of the model by maximizing the accuracy of the model. This process is called parameter tuning or adaptation.

(8) **Support adaptation component:** The support adaptation component compares the estimated cognition data from the human cognition description component with the normative cognition data from the human cognition prescription component. By this comparison, the support adaptation component can detect a possible discrepancy, based on which it can be triggered to do two kinds of interventions: 1) *change* the human-computer interface, filtering or highlighting certain information, or 2) *inform* the task support component to alter its autonomy (remember that these were the two main adaptive support types evaluated in this thesis).

5 Research Methodology

For the development and application of the in the previous section described support system design, also a specific research methodology has been used. This methodology is

graphically depicted in Figure 2 and each phase is further described below.

(a) **Determination of domain and related Human Factors issues:** The developed technology has the purpose of (eventually) being applied in specific domains (s.a. air traffic control, military and crisis response management) in which specific challenging Human Factors issues (s.a. over- and under-trust, confirmation biases, complacency and cognitive under- or over-load) can be (partially) solved by this technology. These domains and the related Human Factors issues have to be determined preferably through identification of real-world problems which preferably society itself mentions as being important to solve and can be (or sometimes can potentially be, in cases of more exploratory research) solved within the current technological design space. Research based on this principle, i.e., generating new or enhanced support systems by increasing insight in the human factor in human-computer interaction, is also called Cognitive Engineering (CE) (Neerincx, 2003). The combined approach of taking into account both Human Factors knowledge as well as the technological design space when developing intelligent support systems is called the CE+ methodology. Where the CE+ is for the development of support systems in general, the methodology described in this chapter (i.e., the introduction) is more specifically for adaptive support systems based on cognitive models, which is the topic of this thesis. This thesis' methodology can therefore be seen as an instantiation of CE+. The CE+ method is further described in Chapter 2. The output of the determination of domain and related Human Factors issues is a set of requirements for both the cognitive models as well as the support system using these cognitive models. From here these requirements propagate through all further methodological phases (including the verification, validation and evaluation phases, explained below).

(b) **Development of informal cognitive models:** Given the requirements from the previous methodological phase, the relevant literature is reviewed to gain knowledge on the underlying problems of the Human Factors issues. Also it is investigated to what extent these issues are already solved in theory or in practice. This should eventually lead to an informal (meaning not executable by a computer) description of the relevant cognitive processes of humans that interact with certain support systems in the chosen domain. One could see this as a blueprint for the further development of the cognitive model as an executable program on a computer. As shown in Figure 1, two types of cognitive models can be developed: descriptive and prescriptive cognitive models. It depends on the specific type of support envisioned what kind of descriptive and prescriptive models need to be developed.

(c) **Psychological experimentation:** For those issues that have not been investigated in sufficient detail for the chosen domain, psychological experiments are designed and performed that help in gaining the lacking knowledge. This eventually leads to a larger or more psychologically valid informal model.

(d) **Formalization of cognitive models:** The above described blueprints of the cognitive models are used for the implementation of executable formal models, i.e.,

models that can be understood and run by a computer. The implementation can be done in different languages: from lower-level programming languages such as JAVA, C#, MATLAB, VB, VBA, Lisp, to higher-level modeling languages such as LEADSTO (Bosse et al., 2007a) and TTL (Bosse et al., 2009a), ACT-R (Anderson and Lebiere, 1998), Soar (Laird et al., 1987), CLARION (Sun, 2002) and 2APL (Dastani, 2008). Both language types have been used for this dissertation.

(e) **Verification of cognitive models:** Verification of cognitive models is important since it is not trivial whether the intended behavior of the model indeed is observed after its implementation. The implementation of the model can have many software engineering-specific challenges, which can lead to deviations from the intended model output. An example of these challenges is for instance to keep the extensive amount of code needed to implement each cognitive model clear from any bugs. In order to check whether the implemented model is internally sound and whether the (intermediary) output of the models is consistent with the expectations from the informally identified required behavioral properties, these behavioral properties are checked against simulation results. The properties are by themselves also first informally described and then formalized in order to let the computer use it to check against the executable formal model. The checking of the properties is done using statistical and verification tools such as the Matlab Statistics Toolbox, SPSS, Statistica, the TTL-checking tool (Bosse et al., 2009a), or checking algorithms are implemented on demand in either one of the previously mentioned programming or modeling languages. Based on the outcomes of the verification simulations, the models are improved incrementally, eventually ending once a certain stopping criterion is reached. For reasons of efficiency, checking can also be done for mere parts or simplifications of the model, so that one can be sure the structure of the model is sound, before one continues to implement the model in a more extensive way.

(f) **Validation and tuning of cognitive models:** This methodological phase is performed to check whether the validity of the output of the cognitive model is sufficient for the envisioned support system. Each validation experiment is basically a comparison between gathered validation data and output of the executable cognitive model. In this phase it is important to verify if the gathered validation data indeed represents the psychological concept which is supposed to be captured in the cognitive model. Both the outcome of the model as well as the validation data are *behavioral consequences* of this psychological concept and not cognition itself. Based on the outcomes of the validation experiments, the models are improved incrementally, eventually ending once a certain stopping criterion is reached. This can be done by hand, by means of altering the code of the model, but it can also be done by means of tuning the parameters of the model given a representative dataset. Properly chosen and tuned parameters are also expected to increase model validity.

(g) **Development of adaptive support system:** In this stage the different (pre- and descriptive) cognitive models are combined within an adaptive support system,

using the support system design shown in Figure 1. A comparison of the implemented descriptive and prescriptive cognitive models leads to interventions that try to guide human cognition from the current to the desired state. The developed adaptive support system has to be designed in such a way that it meets the requirements related to the Human Factors issues in the chosen domain.

(h) **Verification of adaptive support system:** In this methodological phase, it is verified whether the behavior of the support system is as expected. The tools used are similar to the ones used for the verification of cognitive models. The difference in verification lies in the used simulation data: it involves the verification of the reactions of the support system given certain pre-conditions. Based on the outcomes of the verification simulations, the support system is improved incrementally, eventually ending once a certain stopping criterion is reached.

(i) **Evaluation of adaptive support system:** The support system is both objectively as well as subjectively evaluated through experimentation. An object evaluation would for instance be whether human-computer team performance increases compared to non-adaptive support. Subjective evaluation is always done using different questionnaires regarding for instance satisfaction, trust and responsibility. Again, based on the outcomes of the evaluation experiments, the support system is improved incrementally, eventually ending once a certain stopping criterion is reached.

Results regarding the extent to which the above research methodology is applicable is further described in Section 1, as further details are probably better understood after reading the studies in the different chapters.

6 Overview of the Thesis

The format of this thesis is a collection of articles. All chapters are reprints of papers which were published or submitted for publishing elsewhere. References to the respective published papers are given on each chapter title page. The papers are unchanged, with the exception of some layout, grammar and spelling issues. This has three important implications. In the first place, there is overlap between a number of chapters. For example, each chapter contains a specific section in which the modeling approach is introduced again, with special attention to the aspects of the approach that are relevant for the domain in question. Secondly, the fact that most chapters correspond to existing papers implies that each of them can be read in isolation. In other words, this thesis does not have any specific reading order, which is for instance similar to reading the proceedings of a conference. However, those readers that prefer to read the complete thesis are recommended to follow the normal order, starting with Chapter 1 and finishing with Chapter 16. Thirdly, since the different chapters in this thesis are relatively untouched after their publication, the deeper understanding of the subject of this thesis after publishing the newer chapters could not have its impact on the older chapters. The possible

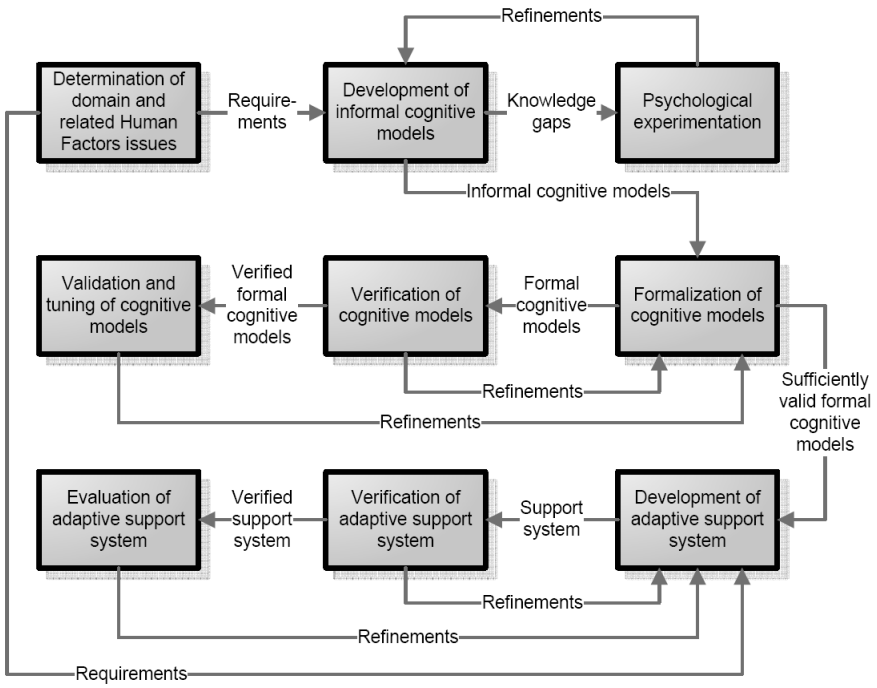


Figure 2: The research methodology used in this thesis.

negative effect of this on the older chapters (or the absence of the positive effect) is limited by an elaborate revisit of all chapters, both in the introduction and conclusion. Also possible newer insights on topics in older chapters have been described in newer chapters, while referring to the older ones. In this way the increased insight over the years is reported as much as possible, while each chapter can still be seen as authentic papers.

In this dissertation two types of cognitive processes (trust and attention) in five different domains are studied. This results in two parts named 'Trust' (Part II) and 'Attention' (Part III) which comprise 13 papers in total (6 about trust and 7 about attention). Also an 'Introduction and Methodology' part (Part I) has been included which comprises the present chapter and a paper discussing more general methodological aspects of the research conducted. The thesis ends with a 'Research Overview and General Discussion' (Part IV) and 'Appendices' part (Part V). The appendices contain the acknowledgments, bibliography, biography and a complete list of published dissertations under the auspices of the SIKS research school.[1]

Below, the in total 16 chapters of only Parts I, II, III and IV are further described.

(I) *Introduction and Methodology*
In Part I the topic of the thesis and the research methodology are introduced.

(1) Chapter 1 (this chapter) introduces the topic of the thesis: (a method for) the development and evaluation of adaptive support for human-computer teams based on the usage of validated computational cognitive models of attention or trust.

(2) In Chapter 2 a general research methodology CE+ is introduced, where special focus is given on the integration of Human Factors research and AI while developing human-computer cooperative systems. The CE+ method can be seen as a generalization of the research method used for the research reported in this dissertation. The general research methodology CE+ represents the combined approach of taking into account both the technological design space (what technology is currently available in order to develop adaptive support?), as well as Human Factors knowledge (what are the limitations of users interacting with the developed adaptive support?). This approach along with several case-studies is further explained in this chapter and is considered an important background paper for understanding the remainder of the thesis.

(II) *Trust*
In Part II the general and specific methodology described in Part I is applied to adaptive support for human-computer teams using cognitive models of *trust*.

(3) In Chapter 3 an abstract experimental task environment is discussed in which one can study human trust and reliance (which is a behavioral implication of trust) dynamics and apply adaptive support using cognitive models of trust. A specific type of adaptive support is also described, using a prescriptive and

[1] A digital version of this thesis along with the source code and material used for the studies reported, can be found at the author's personal website, which is currently `http://www.few.vu.nl/~pp`.

descriptive cognitive model of trust, where tasks are dynamically allocated in a human-computer team in order to improve the team performance. Furthermore, a preliminary experimental design is given using this environment and support type. This design can be used to validate theoretical aspects of the cognitive model and to evaluate the support in terms of human-computer team performance.

(4) As mentioned, the type of support developed and tested in this thesis takes human cognitive processes into account. When such support is able to seamlessly adapt to human cognition, one can think of such support as an augmentation of human cognition. This augmentation can be compared to the way two humans would work and think cooperatively.[2] Chapter 4 argues that it is important to study issues concerning trust when developing systems that are intended to augment cognition. It is important because humans often are mis-calibrated with respect to their trust in support systems that perform certain cognitive tasks autonomously. Several conceptual designs and their design requirements are described of adaptive support systems which make an estimation of the extent of this mis-calibration (under different circumstances). When there is a high expectation of mis-calibration, a system can intervene in three ways: 1) advising the human whether to trust a certain agent or not, but letting the human make the reliance decision (minimal autonomy support), 2) taking over the reliance decision all together and thus taking out the human factor with respect to calibration of trust (maximal autonomy support), or 3) only taking over when it is expected that the human is worse in making reliance decisions, i.e., trust is mis-calibrated (adaptive autonomy support). The possibilities in terms of the application of these ideas are explored and the further development of this concept in terms of the task environment explained in Chapter 3 is also a topic in Chapter 4.

(5) As was argued in Chapter 4, it is important to study Human Factors issues concerning mis-calibration of trust in support systems. The study discussed in Chapter 5 is an example of such research, based on the environment described in Chapter 3. More specifically, the effects of order of advice and the causes of mis-calibration are studied. These findings are potentially applicable for the design of decision aids and training procedures.

(6) In Chapter 6 the in Chapter 4 announced types of support are experimentally evaluated. A combination of laboratory and simulation experiments is done to test whether support of human-computer teams in the second type (maximal autonomy support) indeed leads to an increase of team performance, when comparing to when no such support is applied. Furthermore it is tested whether the third type (adaptive autonomy support) results in an even further improvement of performance. The results are analyzed and further discussed.

[2] One could even say that future versions of such augmented cognition will outperform cooperating human teams. Some influential futurists already think such superiority of computer intelligence is near (Kurzweil, 2005).

(7) Because the previously used experimental environment was of a more abstract type (learning patterns and pushing buttons) and also the human-computer team consisted of mere one human and one computer, another more contextually rich experimental environment has been developed that requires the cooperation between two humans and their supporting computers. This environment is used in Chapter 7 to further explore the possibilities of the application of cognitive models of trust in adaptive support systems for human-computer teams. In the environment a human has to decide on certain courses of action based on several criteria and video footage from Unmanned Areal Vehicles (UAVs). In this chapter the validities of two types of descriptive cognitive models of trust are tested by using validation data retrieved from experiments done using this more contextually rich environment. The first model estimates human trust based on performance feedback of the human independently from the estimated trust in other agents (human or computer), whereas the second model does a relative estimation of human trust. The latter is expected to have higher validity since relativity of trust-calibration is observed in human reliance behavior. Both models are trained on the data retrieved from the experiments, after which the two models are compared to each other. The source code of the trust model tailored for the experiment is also added as an appendix to the chapter.

(8) In Chapter 8 the same environment as used in Chapter 7 is used in order to evaluate two types of adaptive support based on a variant of the second trust model described in Chapter 7 (for descriptive trust) and a variant of the trust model described in Chapter 6 (for prescriptive trust). These two types of adaptive team support have been developed based on the support types described in Chapter 4, namely 1) a minimal autonomy support type (type 1), where the degree of mis-calibration of trust in the self, another human and the computer, is decreased using a graphical representation of the estimation of this degree, and 2) an adaptive autonomy support type (type 3), where the reliance decisions are taken over only when the computer estimates the degree of trust mis-calibration is above a certain threshold. The effects in terms of team performance and satisfaction are discussed for varying human task performance and task difficulty conditions.

(III) *Attention*
Similar as in Part II, in Part III the general and specific methodology described in Part I is applied to adaptive support for human-computer teams using cognitive models of *attention*.

(9) In Chapter 9 the possibilities of adaptive support based on cognitive models of attention are explored. A system is described which is able to reason about the allocation and timing of certain cognitive tasks (also called meta-cognition) requiring visual attention. The domain of naval warfare is introduced, which is composed of complex and dynamic situations in which one has to deal with a large number of tasks in parallel. The envisioned support

system supports naval personnel in dynamically allocating tasks in two introduced task environments, namely 1) an air traffic control environment and 2) a tactical picture compilation task. The envisioned support system and introduced tasks are also used further on in this thesis.

(10) In Chapter 10 the first task introduced in Chapter 9 (the air traffic control task) is used to further explore the means for supporting humans by using cognitive models of attention. The formalization of a descriptive cognitive model of attention is explained and a case study is described in which this model is used to simulate a human subject's attention. This simulation is based on gathered eye-tracker and task execution data from participants executing the task. This simulation of attention is then discussed and formally analyzed. The formal analysis is based on temporal relational specifications for attentional states and for different stages of attentional processes. Five kinds of stages of attentional processes are defined and implemented in logical format which can be automatically checked against the simulated data. The different attentional stages are related to 1) the allocation of attention, 2) attention during the examination of multiple objects, 3) attention during decision making and selection of certain actions to perform, 4) attention during preparation and execution of actions, and finally, 5) attention during action assessment. The automatic detection of the above stages can have several implications for human attention-based adaptive support systems, which are also shortly discussed in this chapter. The source code of the attention model tailored for the experiment and a pre-processing and visualization module are also added as appendices to the chapter.

(11) In Chapter 11, the cognitive model described and analyzed in Chapter 10 is also applied to the second task introduced in Chapter 9. Also a variant of the support system envisioned in Chapter 9 is implemented, of which the results are reported in this chapter. The system is described as an adaptive cooperative agent assisting humans having trouble to allocate attention appropriately. The design is discussed of a component of this adaptive agent, called a Human Attention-Based Task Allocator (HABTA), capable of managing the attention of the human and his assisting agent. The HABTA-component reallocates the human's and agent's focus of attention to tasks or objects based on an estimation of the current human allocation of attention and by comparison of this estimation with certain normative rules. First, an experiment is described which had the purpose of validating the cognitive model of attention. Then an experiment is described which evaluated the HABTA-based support approach. Finally, the results are discussed.

(12) As the developed cognitive model-based support needs to be accurate enough, the used models also need to be valid and robust enough: wrong estimations by the models might even result in worse support, compared to non-adaptive support. For this reason, in Chapter 12, different psychological aspects of the validity of variants of the in Chapter 10 described cognitive model of attention are studied. The effects of task performance and task complexity on

this validity are studied for three different models, namely 1) the gaze-based model, which uses gaze behavior to determine where the subject's attention is, 2) the task-based model, which uses information about the task, and 3) the combined model, which uses both gaze behavior and task information. The models are applied to the second task introduced in Chapter 9 (the tactical picture compilation task), where validation data is gathered by letting participants indicate where their attention is allocated to during different stages of the experiment. These indications are then compared with the estimations of the three models. The results are discussed in the light of possible improvements of the model and applications for the models as fundamental part of adaptive support systems.

(13) In Chapter 13 a new task environment is introduced. The task involves the identification of incoming flying objects and deciding whether to shoot the object or allowing it to maintain its course. An improved variant of the model described in Chapter 10 is tailored to this new task. Similar as in Chapter 7 in Part II about trust, exploring new applications is expected to lead to better understanding of the scalability and the further possibilities of using the methodology of applying cognitive models in adaptive support systems. Furthermore it is investigated whether it is possible to improve the validity of the model by personalizing the models to each participant given the data of these participants from different experiments. Similar as in Chapter 12, it is stressed that cognitive model-based adaptive support systems can benefit from this increased validity. The idea of personalization is based on the fact that different characteristics might determine the optimal parameter values in the used cognitive models. A Simulated Annealing (SA)- and Area under the Curve (AUC)-approach is used to find the optimal validities of either the personalized or the fixed models.

(14) In Chapter 14 a more elaborate version of a variant of the in Chapter 11 discussed support system is studied. This chapter is about the architecture of a supporting agent that is able to manipulate the visual attention of a human. Like in Chapter 11, this agent model is applied to the second task introduced in Chapter 9 (the tactical picture compilation task). The agent model consists of four formalized sub-models, namely 1) a dynamic attention model based on the model studied in Chapter 10, 2) a model for beliefs about attention, 3) a model to determine the discrepancy between the estimated subject's attention (descriptive attention) and normative attention (prescriptive attention) and 4) a decision model for attention adjustment (i.e., the 'manipulation' of the subject's attention). A large amount of data has been gathered during different experiments with the described agent. This data is used to formally analyze and verify the support system, using automated checking tools. The results of this analysis and verification is discussed in the light of possible future improvements of the support system. The source code of the attention model tailored for the experiment is also added as an appendix to the chapter.

(15) In Chapter 15 three variants of the in Chapter 14 described support system are

evaluated, using the second task environment introduced in Chapter 9. The differences in benefits of these three types are investigated in terms of the resulting team performance (similar as in Chapter 8), trust, understandability and responsibility. The types of adaptive support are different with respect to their level of conservativeness. In the fixed support condition, the participant's attention is drawn by highlighting contacts which are automatically classified by the support system. In the liberal adaptive support condition, attention is drawn to contacts that are most likely to be incorrectly classified by the participant. And finally, in the conservative condition, attention is drawn to these same contacts, but only when participants are not expected to be attending to them. It is expected that adaptive decision support reduces inappropriate reliance on advice from a support system. The results are discussed in the light of new possible improvements for attention allocation support systems in complex visually aided computer tasks.

(IV) *Discussion*

(16) In Chapter 16 a summary of the conclusions of this dissertation is discussed together with related and possible future work.

Chapter 2

Integrating Human Factors and Artificial Intelligence in the Development of Human-Machine Cooperation

This chapter appeared as (van Maanen et al., 2005).

Integrating Human Factors and Artificial Intelligence in the Development of Human-Machine Cooperation

Peter-Paul van Maanen[*‡], Jasper Lindenberg[*] and Mark A. Neerincx[*†]

[*] Department of Information Processing, TNO Human Factors
P.O. Box 23, 3769 ZG Soesterberg, The Netherlands
Email: {peter-paul.vanmaanen, jasper.lindenberg, mark.neerincx}@tno.nl
[†] Man-Machine Interaction Group, Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands
[‡] Department of Artificial Intelligence, Vrije Universiteit Amsterdam
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands

***Abstract*—Increasing machine intelligence leads to a shift from a mere interactive to a much more complex cooperative human-machine relation requiring a multidisciplinary development approach. This paper presents a generic multidisciplinary cognitive engineering method CE+ for the integration of human factors and artificial intelligence in the development of human-machine cooperation. Four case-studies are presented which contain a description of the developed human-machine cooperation and the adjusted CE+ method used. For each case-study the method supported research and development activities in such a way that sound knowledge bases, methodologies, and user interfaces for human-machine cooperation could be established. However, the method always needed to be tailored to the specific goals and circumstances, such as the available time, novelty, and required integration.**

***Index Terms*—Human-machine cooperation, integrated system design, cognitive engineering, human factors, intelligent user-interfaces.**

1 Introduction

Living, travel and working environments contain a growing number of networked information compilations and electronic services (e.g., healthcare and security services), which are accessible to an increasing number of diverse user groups. In current human-computer interaction (HCI) research, personalization, adaptive interfaces and electronic assistants are proposed to enable easy access to the proliferating functions and services in such environments for both the consumer and professional domain (e.g., Aarts et al., 2001; Abowd et al., 2002; Satyanarayanan, 2001). The increasing intelligence of machines leads to a shift from HCI to human-machine *cooperation* (HMC) (Hoc, 2001). Future machines will either be designed to cooperate, or designed to learn how to cooperate, with humans. They will be able to assess and adapt to human goals (Castelfranchi, 1998). It was only first mentioned in (Hollnagel and Woods, 1983) that there is a growing need for humans and machines to comprehend each other's reasoning and behavior. And since the last decade or so, one is beginning to realise that exactly this really requires researchers with different backgrounds to believe in a more multidisciplinary approach.

For HMC the aim is to customize support by accommodating individual user characteristics, tasks and contexts in order to establish HMC in which the computer provides the "right" information and functionality at the "right" time and in the "right" way (Fischer, 2001).

The customization that one encounters today at work, during travel or at home is rather limited, appearing as static user interfaces with simple or "local" adaptations (Schneider-Hufschmidt et al., 1993; Aarts et al., 2003). The possibilities for HMC are extensive, however knowledge is lacking on both the specific human factors (HF), the artificial intelligence (AI) prospects and on ways of successfully integrating both HF and AI during development. This paper focuses on the latter, the integration of HF and AI during research and development (R&D) of HMC. An extensive and diverse set of HF

methods and tools are distinguished and proposed for the design of tasks and user interfaces, for instance from the perspective of (cognitive) task analysis (e.g., Kirwan and Ainsworth, 1992; Schraagen et al., 2000; Hollnagel, 2003), HCI (e.g., Helander et al., 1997; Jacko and Sears, 2003) and usability engineering (e.g., Mayhew, 1999; Maguire, 2001; Rosson and Carroll, 2001). Furthermore, there is an extensive and diverse set of guidelines and standards for HCI in general (e.g., Bevan, 2001), and for specific application domains (e.g., NASA standards, 1998). A major challenge for the development of complex and dynamic human-machine systems — such as industrial process control, aerospace and traffic control — is to develop HMC and realize concrete design practices in the near future. A suitable candidate for this activity is cognitive engineering with its roots in both principal contributors HF and AI. Other available development methods are too heavily focused on their own origin (human or technology), and have a blind spot for the other domain. Methods focused on integration such as MUSE (Lim and Long, 1995) or even ISO 13407 are not well suited for innovation. An extended generic cognitive engineering method CE+ is presented and four case-studies illustrate the use of this method and the required adjustments based on specific project requirements and circumstances.

2 The Cognitive Engineering Method CE+

Cognitive engineering (CE) approaches originated in the 1980s to improve computer-supported task performance (e.g., Rasmussen, 1986; Norman, 1986) and emerged from the fields of cognitive science and AI. CE aims at generating new or enhanced HCI by increasing insight in the cognitive factors of human performance (Neerincx, 2003). Furthermore, CE guides the iterative process of development in which an artifact is specified in more-and-more detail and specifications are assessed more or less regularly to refine the specification, to test it, and to adjust or extend it. The original CE methodology was extended with an explicit technology input thus creating the CE+ method. This extension was primarily made because of two reasons. First, the technological design space sets a focus in the process of specification and generation of ideas. Second, the reciprocal effects of technology and HF are made explicit and are integrated in the developent process. In Figure 1 the development process of the extended method CE+ is shown. The HF knowledge provides relevant expertise (i.e., guidelines and support concepts) and techniques for the specification and assessment of HMC. The technological design space sets the technological and operational requirements for HMC. In the specification both the guidelines and the technological design space must be addressed concurrently. In the assessment it is checked whether the specifications agree with these guidelines and the technological design space. An assessment will provide qualitative or quantitative results in terms of effectiveness, efficiency, satisfaction and user experience which are used to refine, adjust or extend the specification. Eventually, the process of iteration stops when the assessment shows that the HMC satisfies all requirements (Neerincx et al., 1999). The above thus suggests dynamic integration of knowledge into the design process rather than *a priori* specification of guidelines.

3 Case-studies

3.1 Personal Assistant for onLine Services

The Personal Assistant for onLine Services (PALS) project was aimed at substantially improving the user experience of mobile internet services (Lindenberg et al., 2003). It focused on a generic solution: a personal assistant, which attunes the interaction to the momentary user needs and use context (e.g., adjusting the information, presentation and navigation support to the current context, device and interests of the user).

The PALS project was carried out using CE+. The method was adjusted to fit the specific needs of the PALS project. The goal of the project was not only to realize an effective and efficient PALS but also to generate fundamental HF and AI knowledge. Therefore, three research lines can be distinguished within the adjusted method for the PALS project (Figure 2):

1) PALS creation: using a cognitive engineering approach.
2) Basic HF research: extending the HF knowledge base.
3) Basic technological research: extending the AI knowledge and engineering base

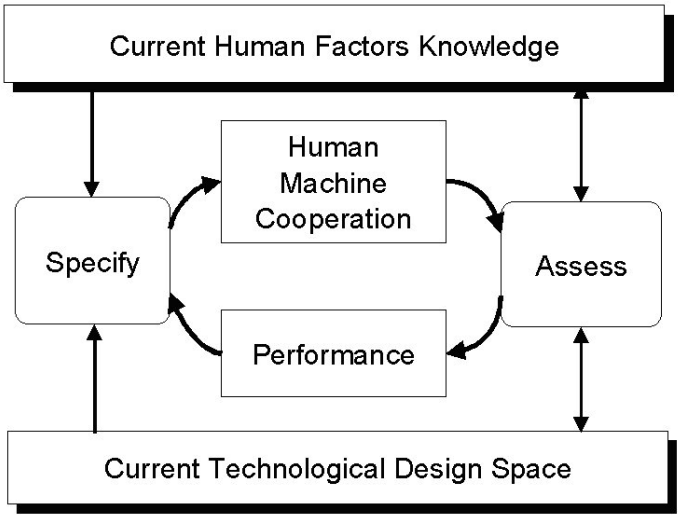


Figure 1. The development process of the CE+ method.

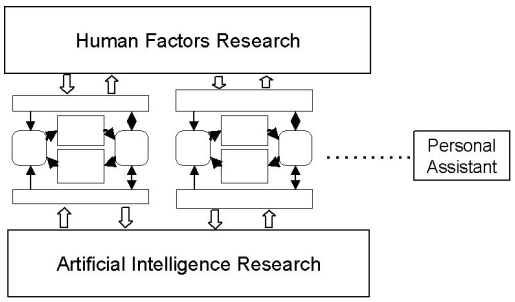


Figure 2. The CE+ development process in the PALS project.

The first research line focused on the actual realisation of a PALS demonstrator guided by the cognitive engineering process (Figure 3). In different stages knowledge and/or technology was needed that was not available at that time. This knowledge was developed within the two, discipline focused, research lines of PALS, both enabling the realization of an effective PALS and extending the HF and AI knowledge base. For example, the influence of attention on mobile user interaction, and the AI techniques to attune the interaction to the users attentional state. These issues were examined by developing a rule-based in-car system that predicted the momentary mental load caused by the driving task and attuned the dialogue accordingly to prevent overload. In addition to the CE+ generated questions that "fed" the basic research, autonomous processes within the basic research line "fed" the CE+ process by providing new interaction concepts. The specific circumstances of this project such as the combination of fundamental research with prototype development, the relatively long running time, and the physical distance between the participating partners gave rise to the specific method that was used. The integration of HF and AI technology in PALS resulted for example in a Point of Return indicator, an Interactive Suspension Point and a Tailored Information View, based on mining and (graph) modeling of user behaviour data and the identification of HF bottlenecks in mobile environments.

3.2 Context Aware Communication Terminal and USer

The Context Aware Communication Terminal and USer (CACTUS) project aimed at researching technological and usability aspects of human-machine and machine-network interaction with personalized, intelligent and context-aware wearable devices in

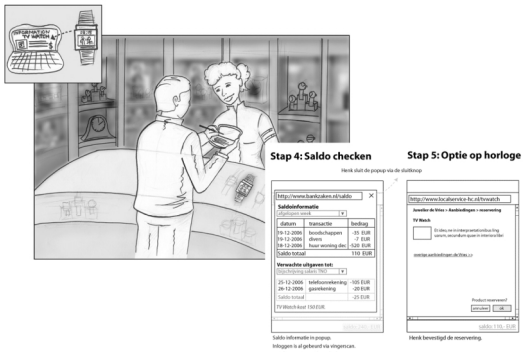


Figure 3. Scenario based design was used to specify the interaction.

ad-hoc wireless environments such as the future home, office, or university campus. In this paper we will focus on a part of the project which was concerned with the selection and identification of agents in an ubiquitous computing environment. Future AI and HF problems were identified by a technology assessment early on in the project. It turned out that current techniques for identification and selection of agents in ubicom environments were not scalable leading to all types of HF and AI performance problems. This instigated a research program containing both an AI and HF challenge: create a scalable decentralized agent system which enables users to identify and select the best service to obtain their goals. The CE+ method described in the previous section contained two separate, domain specific, research lines which are integrated by a third development line. That particular set-up was not suited for CACTUS because of the limited amount of time that was available and the strict interactions between the AI and HF challenge. Therefore, both domains were studied in an integrated manner. A realistic technological solution for the predicted HF problems was conceived and implemented within a limited environment. This technology enabled the user to simply express his goal, in a decentralized manner each agent decided whether or not it was capable. The most capable agents would rise to the surface and offer their services to the user. Early on in the development the technology was empirically tested with a realistic mock-up in which the actual behavior of the technology was simulated by a human operator (Wizard of Oz) (see (Lindenberg et al., 2007) and Figure 4). Because experiment showed a significant increase in user performance the decision was made to extend the implementation to a larger environment. The data that was gained during the experiment was actually used as excellent training data for the final implementation providing another argument for joint HF and AI research (Pasman and Lindenberg, 2006).

Figure 4. Early empirical testing of ubicomp agent archtecture with end-users.

The development process of the method that was used is shown in Figure 5. The assessment in the final iteration of the development showed that both AI and HF challenges for agent selection in a dynamic, large and ad-hoc agent environment were met.

3.3 Situated Usability Engineering for Interactive Task Environments

Intelligent operation support is crucial for human-machine performance in space laboratories. A tool kit for "Situated Usability engineering for Interactive Task Environments" (SUITE) was developed to guide the design of such operation support, in order to harmonize the activities of diverse stakeholders who implement various applications (platform systems and so-called payloads), apply specific design techniques and focus on the development of either (intelligent) task support or displays (Neerincx et al., 2004). SUITE consists of a usability engineering handbook that provides context- and user-tailored views on the recommended HF method, guidelines and best practices. Furthermore, it provides a generic task support and dialogue framework, called Supporting Crew OPErations (SCOPE), as both an implementation of these methods and guidelines, and an instance of current interaction and

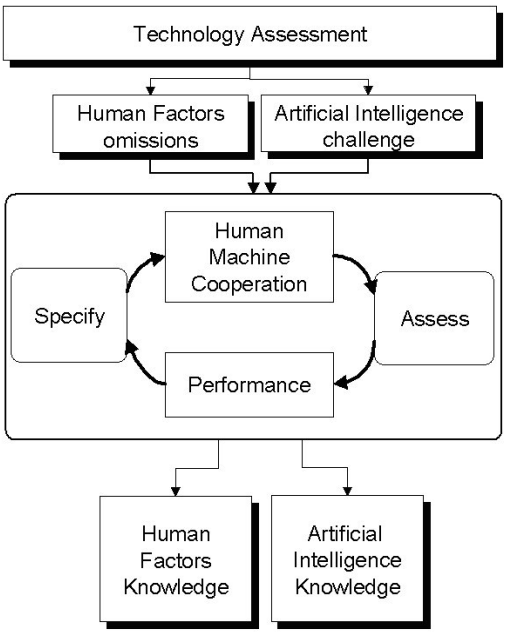


Figure 5. The CE+ development process in the CACTUS project.

AI technology for HMC. This framework defines a common multi-modal interaction with a system, including the integrated provision of context-specific task support for nominal and off-nominal situations. Furthermore, SCOPE detects system failures, guides the isolation of the root causes of failures, and presents the relevant repair procedures in textual, graphical and multimedia formats (see Bos et al., 2004). The diagnosis is a joint astronaut-SCOPE activity. Taken HF into account, the tasks of the human and machine actors, and their interactions, were specified and assessed as a joint activity. When needed, SCOPE asks the astronaut to perform additional measurements in order to help resolve uncertainties, ambiguities or conflicts in the current machine status model. SCOPE will ask the user to supply values to input variables it has no sensors for measuring by itself. Each new question is chosen on the basis of an evaluation function that can incorporate both a cost factor (choose the variable with the lowest cost) and a usefulness factor (choose the variable that will provide the largest amount of new information to the diagnosis engine). After each answer, the diagnosis re-evaluates the possible fault modes of the system on the basis of the additional values (and new samples for the ones that can be measured). As soon as SCOPE has determined the likeliest health state(s) of the system with sufficient probability, it presents these states to the user, possibly with suggestions for appropriate repair procedures that can be added to the todo list and executed. As soon as the machine has been repaired, SCOPE will detect and reflect this.

SCOPE was applied for the Cardiopres, a portable payload for medical experimentation (see Figure 6). In the evaluations of the SCOPE system for the

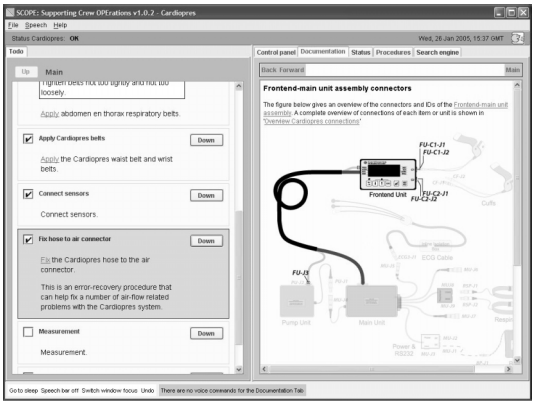


Figure 6. SCOPE showing a successful completion of a diagnosis process (green status bar at top), procedure generation (left), and reference documentation (right).

Cardiopres, the user interface and AI-based task support functions proved to be effective, efficient and easy to learn, and astronauts were very satisfied with the system (Neerincx et al., 2004).

The development of the SUITE tool kit is an iterative process in itself, and new experiences with its application (e.g., currently for a new payload) will improve it. Currently, the SCOPE framework is being applied for the development of an intelligent user interface for the Pulmonary Function System (PFS) payload. Its task support functions will be improved to deal with dependencies of actions with each other and the usage context. Assessments will help to establish adequate performance and user experience of this component (see Figure 7). In

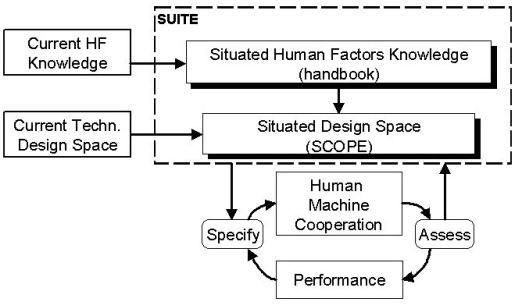


Figure 7. The CE+ development process implemented as design guidance for intelligent user interfaces of space missions.

general, the SUITE toolkit reduces the time and cost of development efforts, whereas it improves the usability of intelligent interfaces. Embedded in a HF engineering process, user interfaces and the underlying AI methods are systematically and coherently specified, implemented and assessed from early development phases on, which is in itself efficient and prevents the need for late harmonisation efforts between user requirements and technological constraints.

3.4 Human-Machine Task Integration

In contrast with the previous subsections, this last subsection describes the analysis of the CE+ applicability on an *ongoing* programme. Resultingly, this case-study is based on expectation resulting from previous cases rather than on plain results. In the "Human-Machine Task Integration" (HMTI) programme human-machine task integration concepts are developed, tested, and evaluated in order to come to a recommended methodology to considerably improve performance with respect to HMC on future navy platforms, based on non-fixed HF and AI knowledge. Those scenarios are considered that contain dynamic, unreliable, and ambiguous environments, and systems that operate under time pressure, and with less resources (e.g., manning). Notably, these situational aspects are nowadays repeatedly mentioned as typical for what we can expect already in the near future. Exactly these (should) motivate governments to fund research on the integration of tasks through HMC systems. In Figure 8 an implementation of early HMTI for future navy platforms is shown.

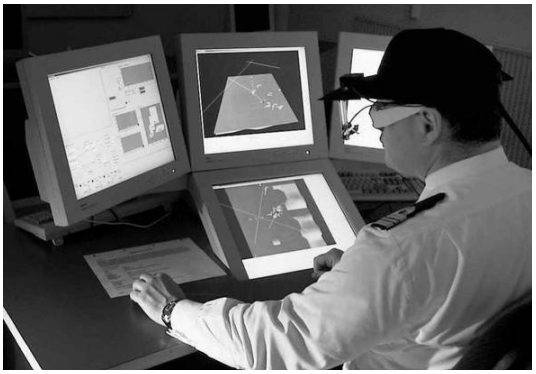


Figure 8. Early HMTI for future navy platforms.

Adaptive HMC (AHMC) systems attempt to adapt to the human-machine relation complexity. AHMC is an approach to design where tasks are dynamically allocated over time between humans and machines for the purpose of optimizing overall system performance. The apparent underpinnings of AHMC consisted of, among others, loss of expertise, automation-induced complacency, over- and undertrust, and loss of adaptivity (e.g., Parasuraman and Riley, 1997; Moray, 1997; Rouse, 1994). To overcome these problems, future HMC systems should detect and adapt to those situations that cause them.

Research in HMTI can be divided into two main foci. The first focus is on when specific types of cooperation should be changed (triggering or invocation strategy), and the second one is on what and how it should be changed (response or allocation strategy). Guided by HF and AI research, these together span the whole AHMC system design space. In general, invocation strategies are based on the characteristics of, and changes in, the human-machine system, its environment, and estimated future performance models. After this, chosen allocation strategies cause new characteristics and changes. In complex, ambiguous, and dynamic environments this choice must be made *a posteriori*, i.e., real-time.

What can be determined *a priori*, i.e., during design, is everything that constitutes the design purpose, such as the choice that the allocation

strategy is based on a left-over, economic, or comparison method (Rouse, 2001). Another choice a designer can make is what type of support HMC should provide and more specifically what type of sharing of control. The type of sharing of control designates in what way agents (machine or human) cooperate to achieve the system's goals. There are three sharing of control strategies, namely extension, relief, and partitioning (Sheridan, 1992). It is clear that these control strategies require different intelligence of the cooperating partner. In many cases extension simply requires precompiled tools, whereas partitioning sometimes needs an agent to be even more intelligent than the subject. Also, in partitioning cooperation will require the cooperating agents to perform additional meta-operations (Hoc, 2001), which are to be relieved by means of a well-equipped AHMC design methodology.

Given the research aims of the area of integrated system design, we can definitely claim that there is still a lot of work to be done. With respect to AHMC design, in spite of its popularity in the past decades, there is very little formal research to be found that can improve the design of large complex systems (e.g., Fuld, 1993; Scallen and Hancock, 2001). There are few usable models for predicting the dynamics of human or machine state, performance, and environment. Therefore more research on its theoretical framework is needed. Models need to be developed that can closely predict situation awareness, vigilance, mode awareness, automation-induced complacency, mental load, boredom, emotion, skill, experience, stress, self-confidence, trust, and commitment (to name but a few), and determine their characteristics in terms of for example demand for transparency, machine autonomy, responsibility, "out of the loop"-ness, task switching, and delegation strategy. These models may depend on specific task, environment, machine, user, or organization characteristics. Further research also applies to the *formalization*, *verification*, and *validation* of these models. This is for the reason that well-balanced models should be consistent when combined, refrain from under- as well as overfitting instances of reality, and result in implementations that are *application valid*. The latter may imply theories with low construct validity, as is discussed recently in (Campbell and Bolton, 2004).

As an important first result of the HMTI programme the above clearly implies that HF and AI research are thoroughly intertwined into the HMC system design process. This suggests the applicability of the CE+ method and this is why the HMTI programme has adopted it. In Figure 9 the proposed CE+ development process is shown. The initial knowledge helps in setting system constraints and show important gaps that indicate a need for further research. After an experimental phase, results are reflected upon the initial theory by means of comparing desired and resulting perfomance. This gains new knowledge and new concepts are further studied. Important here is that after several of such

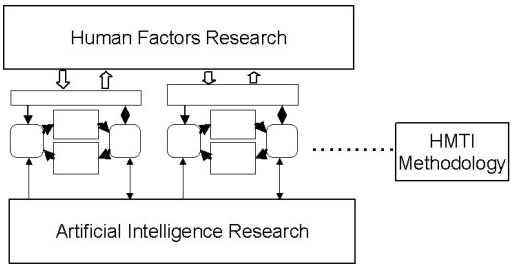


Figure 9. The proposed CE+ development process in the HMTI programme.

iterations the resulting methodology is not technology driven, but rather realistic for future navy platform scenarios in a generic sense. This is why the current technological design space is missing in this diagram and AI research is directly integrated in the specification and assessment processes. Indeed, few gaps will be identified when not using any CE+ method. Eventually the resulting AHMC system design methodology will be useable without first going through phases of trial and error.

4 Conclusion

Increasing machine intelligence leads to a shift from a mere interactive to a much more complex cooperative human-machine relation. Exactly this really requires researchers and engineers to believe in a more multidisciplinary approach. This paper stressed validity and therefore usability of a generic multidisciplinary cognitive engineering method CE+ in human-machine cooperation system design by

means of four case-studies. For each case-study the method supported research and development activities in such a way that sound knowledge bases and user interfaces for human-machine cooperation could be established. This was done for example by deriving artificial intelligence and human factors requirements for the attention driven dialogue (PALS), for the hypotheses generation, approval or falsification (SUITE, (Bos et al., 2004)), for adaptivity of automated decision support (HMTI), and agent selection in large ad-hoc environments (CACTUS, (Pasman and Lindenberg, 2006; Pasman, 2004)). However, the method always needed to be tailored to the specific goals and circumstances, such as the available time, novelty, and required integration. We can conclude that due to the complexity of system design processes, their success depends upon integration of human factors and artificial intelligence research early on in the development process.

Acknowledgments

The PALS project is partly funded by the IOP MMI program of SenterNovem. The Human-Machine Task Integration programme is funded by the Dutch Ministry of Defense (programme nr. V206). The CACTUS project was supported by the Towards Freeband Communication Impulse of the CIC programme of the Ministry of Economic Affairs in The Netherlands. The SUITE tool-kit was developed for the European Space Agency (contract C16472/02/NL/JA) in cooperation with Science & Technology (The Netherlands). Gratitude goes to Jan Maarten Schraagen and Egon van den Broek for their helpful comments.

References

Aarts, E., Collier, R., van Loenen, E., and de Ruyter, B. (2003). *Ambient Intelligence: EUSAI 2003. Lecture Notes in Computer Science*. Springer, Berlin.

Aarts, E., Harwig, R., and Schuurman, M. (2001). Ambient intelligence. *The Invisible Future*, pages 235–250.

Abowd, G. D., Mynatt, E. D., and Rodden, T. (2002). The human experience. *Pervasive Computing*, pages 48–57.

Bevan, N. (2001). International standards for hci and usability. *International Journal of Human-Computer Studies*, 55:533–552.

Bos, A., Breebaart, L., Neerincx, M. A., and Wolff, M. (2004). SCOPE: An intelligent maintenance system for supporting crew operations. In *Proceedings of IEEE Autotestcon 2004*, pages 497–503.

Campbell, G. E. and Bolton, A. E. (2004). HBR validation: Integrating lessons learned from multiple academic disciplines, applied communities and the AMBR project. In Pew, R. W. and Gluck, K., editors, *Modeling Human Behavior with Integrated Cognitive Architectures: Comparison, Evaluation, and Validation*. Lawrence Erlbaum. (in press).

Castelfranchi, C. (1998). Modelling social action for agents. *Artificial Intelligence*, 103:157–182.

Fischer, G. (2001). User modeling in human-computer interaction. *User Modeling and User-Adapted Interaction*, 11:65–68.

Fuld, R. B. (1993). The fiction of function allocation. *Ergonomics in Design*, 1:20–24.

Helander, M., Landauer, T., and Prabhu, P. (1997). *Handbook of human-computer interaction*. North-Holland, Amsterdam.

Hoc, J.-M. (2001). Towards a cognitive approach to human-machine cooperation in dynamic situations. *International Journal of Human-Computer Studies*, 54(4):509–540.

Hollnagel, E. (2003). *Handbook of cognitive task design*. Lawrence Erlbaum Associates, Mahwah, NJ.

Hollnagel, E. and Woods, D. D. (1983). Cognitive systems engineering: New wine in new bottles. *International Journal of Man-Machine Studies*, 18(6):583–600.

Jacko, T. A. and Sears, A. (2003). *The Human-Computer Interaction Handbook: Fundamentals, evolving technologies and emerging applications*. Lawrence Erlbaum Associates, Mahwah, NJ.

Kirwan, B. and Ainsworth, L. K. (1992). *A guide to task analysis*. Taylor & Francis, London, UK/Washington, DC.

Lim, K. Y. and Long, J. B. (1995). *The Muse Method for Usability Engineering*. Cambridge University Press, UK.

Lindenberg, J., Nagata, S. F., and Neerincx, M. A.

(2003). Personal assistant for online services: Addressing human factors. In Harris, D., Duffy, V., Smith, M., and Stephanides, C., editors, *Human-Centred Computing: Cognitive, Social and Ergonomic Aspects*, pages 497–501, London. Lawrence Erlbaum Associates.

Lindenberg, J., Pasman, W., Kranenborg, K., Stegeman, J., and Neerincx, M. A. (2007). Improving service matching and selection in ubiquitous computing environments: A user study. *Personal and Ubiquitous Computing*, 11(1):59–68.

Maguire, M. (2001). Methods to support human-centered design. *International Journal of Human-Computer Studies*, 55:587–634.

Mayhew, D. J. (1999). *The usability engineering lifecycle: A practitioner's handbook for user interface design*. Morgan Kaufman, San Fransisco, CA.

Moray, N. (1997). Human factors in process control. *Handbook of Human Factors and Ergonomics*, pages 1944–1971.

NASA standards (1998). Payload display developers guide. annex 6 of the international space station, united states payload operations, data file management plan. *Appendix H - Payload displays of the document DGCS (Display and Graphics Commonality Standards)*,International Space Station program document (SSP) 530313;*Appendix I - Payload displays of the document DGCS (Display and Graphics Commonality Standards)*,International Space Station program document (SSP) 530313, Revised Draft.

Neerincx, M. A. (2003). Cognitive task load design: model, methods and examples. In Hollnagel, E., editor, *Handbook of Cognitive Task Design*, chapter 13, page 283305. Lawrence Erlbaum Associates, Mahwah, NJ.

Neerincx, M. A., Cremers, A. H. M., Bos, A., and Ruijsendaal, M. (2004). A tool kit for the design of crew procedures and user interfaces in space stations. Technical Report TM-04-C026, TNO Human Factors, Soesterberg, The Netherlands.

Neerincx, M. A., Pemberton, S., and Lindenberg, J. (1999). U-wish web usabilty: methods, guidelines and support interfaces. Technical Report TM-99-D005, TNO Human Factors, Soesterberg, The Netherlands.

Norman, D. A. (1986). Cognitive engineering. In Norman, D. A. and Draper, S. W., editors, *User-Centered System Design: New perspectives on human-computer interaction*. Erlbaum, Hillsdale, NJ.

Parasuraman, R. and Riley, V. A. (1997). Humans and automation: Use, misuse, disuse, abuse. *Human Factors*, 39:230–253.

Pasman, W. (2004). Organizing ad hoc agents for human-agent service matching. In Harris, D., Duffy, V., Smith, M., and Stephanides, C., editors, *Proceedings of Mobiquitous 2004*, pages 278–287, Boston, MA.

Pasman, W. and Lindenberg, J. (2006). Human-agent service matching using natural language queries: System test and training. *Personal and Ubiquitous Computing*, 10(6):393–399.

Rasmussen, J. (1986). *Information processing and human-machine interaction: an approach to cognitive engineering*. Elsevier, Amsterdam, The Netherlands.

Rosson, M. B. and Carroll, J. M. (2001). *Usability engineering: Scenario-based development of human-computer interaction*. Morgan Kaufman, San Fransisco, CA.

Rouse, W. B. (1994). Twenty years of adaptive aiding: Origins of the concept and lessons learned. *Human Performance in Automated Systems: Current Research and Trends*, pages 28–32.

Rouse, W. B. (2001). *Design for success: A human centered approach to designing successful products and systems*. Wiley, New York.

Satyanarayanan, M. (2001). Pervasive computing: Vision and challenges. *IEEE Personal Communications*, pages 10–17.

Scallen, S. F. and Hancock, P. A. (2001). Implementing adaptive function allocation. *International Journal of Aviation Psychology*, 11:197–221.

Schneider-Hufschmidt, M., Kühme, T., and Malinowski, U. (1993). Adaptive user interfaces: principles and practices. *Human Factors in Information Technology*.

Schraagen, J. M. C., Chipman, S. E., and Shalin, V. L. (2000). *Cognitive Task Analysis*. Lawrence Erlbaum Associates, Mahwah, NJ.

Sheridan, T. B. (1992). *Telerobotics, Automation, and Human Supervisory Control*. MIT Press, Cambridge, MA.

Part II

Trust

Chapter 3

Towards Task Allocation Decision Support by means of Cognitive Modeling of Trust

This chapter appeared as (van Maanen and van Dongen, 2005a). Also an extended abstract (van Maanen and van Dongen, 2005b) appeared of this chapter.

Towards Task Allocation Decision Support by means of Cognitive Modeling of Trust

Peter-Paul van Maanen*† and Kees van Dongen*

* Department Human in Command, TNO Human Factors
P.O. Box 23, 3769 ZG Soesterberg, The Netherlands
Email: {peter-paul.vanmaanen, kees.vandongen}@tno.nl
† Department of Artificial Intelligence, Vrije Universiteit Amsterdam
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands

***Abstract*—An important issue in research on human-machine cooperation concerns how tasks should be dynamically allocated within a human-machine team in order to improve team performance. The ability to support humans in task allocation decision making requires a thorough understanding of its underlying cognitive processes, and that of relative trust more specifically. This paper presents a computational agent-based model of these cognitive processes and proposes an experiment design that can be used to validate theoretical aspects of this model.**

1 Introduction

The increasing intelligence of machines leads to a shift from HCI to human-machine *cooperation* research (Hoc, 2001). Problems arise when small human-machine teams try to cooperate on a cognitive level. A goal in human-machine cooperation research is to solve these problems. Optimizing performance of the human-machine team is not likely to be gained by improving human-alone or machine-alone performances. It is important that cooperative tasks within the team, and more specifically the dynamic allocation of tasks, are improved as well. This requires an understanding of the cognitive processes underlying task allocation decisions. A useful cognitive theory of task allocation decision making should represent those attributes and their relations that are considered in making decisions on task allocation. A validated model can subsequently be used by decision support systems to support 1) the acquisition of information concerning these attributes, 2) the analysis and integration of this information, 3) the selection of appropriate changes in task allocation, and 4) the execution of these actions (Parasuraman et al., 2000).

Although there has recently been an increase in human factors research concerning trust and automation reliance (Dzindolet et al., 2003; Lee and See, 2004; Lee and Moray, 1992, 1994; Parasuraman and Riley, 1997; Gao and Lee, 2004), few attempts have been undertaken to formalize the cognitive processes underlying task allocation decisions (Fuld, 1993; Scallen and Hancock, 2001). Therefore more research on its theoretical framework is needed. In the AI and sociology community research on the formalization of trust and delegation decisions is present (e.g., Falcone and Castelfranchi, 2001; Gambetta, 1990), but not specifically with respect to dynamic decision making in human-machine cooperation.

The present research attempts to bridge this gap between human factors and AI research by developing a computational model of task allocation decision making that can be used in further understanding and supporting human-machine cooperation. It is work in progress. First, the theoretical aspects of task allocation decision making are introduced. Second, a formal cognitive model is defined. And third, based on this model, an experimental environment is described that can be used to validate the theoretical aspects.

2 Cognitive Theory

As in (Falcone and Castelfranchi, 2001), in this paper the term *trust* is used to refer to a mental state, a belief of a cognitive agent i about the achievement of a desired goal through another agent

j or through agent i itself. In trusting agent j, agent i has, to some level, a positive expectation that agent j's actions will achieve the goal that agent i desires. Agent i's expectation of j's performance is calibrated by direct experience with i's performance. Trust is dynamic, but it does not simply increase and decrease with positive and negative experiences. How trust changes by successes and failures, for one, depends on how increases and decreases in performance are interpreted and causally attributed (Falcone and Castelfranchi, 2004; Lee and See, 2004). Trust is more than other concepts subject to error. One type of error is that humans tend to overestimate their own performance. Humans, for instance overestimate the number of tasks they can complete in a given period of time (Buehler et al., 1994). Another type of error occurs when humans form expectations about the performance of automation. It is found, for instance, that humans have a bias toward automation (Dijkstra et al., 1998; Dzindolet et al., 2002).

There are also indirect sources of knowledge about performance. Reputation and gossip, for instance, enable agents to develop trust without any direct experience. In the context of trust in automation, response times to warnings tend to increase when false alarms occur. This effect was counteracted by gossip that suggested that the rate of false alarms was lower than it actually was (Bliss et al., 1995). Trust can also be based on analogical judgments, i.e., judgment about the trustworthiness of a category rather than on the actual performance of one of its presumed members. Although not always recognized by analytical approaches to trust, it should be noted that humans are cognitive misers and try to save the effort that is required in deliberation. In naturalistic setting it is observed that decision makers seldom engage in extensive information acquisition, conscious calculations or in an exhaustive comparison of alternatives (Klein, 1998). In these multi-tasking environments automatic processes play a substantial role in attributional activities, with many aspects of causal reasoning occurring outside conscious awareness. In (Miller, 2002) for instance it is suggested that computer etiquette may have an important influence on human-machine cooperation. Etiquette may influence trust because category membership associated with adherence to a particular etiquette helps people to infer how automation will perform.

Many theories in the human factors literature about the cognitive processes underlying task allocation decisions include a notion of relative trust, i.e., differences of trust in two agents. Empirical results from human factors experiments show that as the trust in machine performance is significantly higher than trust in own performance, humans intend to allocate tasks to the machine, and when the reverse is true, humans prefer to allocate tasks to themselves (Lee and Moray, 1992; Moray et al., 2000; Dzindolet et al., 2000; De Vries et al., 2003). Theories on these results describe factors that affect trust in machine performance, such as machine performance reliability and error costs. Factors that affect trust in own performance are for instance task difficulty, skill, cognitive biases and the effects of social and motivational processes (Dzindolet et al., 2000).

Trust is distinguished from the decision to allocate a task to an agent or rely on an agent. The term *task allocation decision* is used to refer to the decision to rely on an agent's goal-directed actions to achieve a desired goal. One might argue that an agent is more likely to rely on another agent when its workload is high compared to when it is moderate or low. In (Parasuraman and Riley, 1997), however, it is pointed out that the relation between workload and the reliance decision has not been empirically validated and it is suggested that this relation is obscured by individual differences. In (Kirlik, 1993) it is shown that humans do not simply allocate tasks to automation so as to free up mental resources for concurrent tasks. It has been hypothesized that reliance decisions are not only influenced by individual differences, such as skill on the task or costs of delaying concurrent tasks, but also by the effort or time needed to engage automation. It is expected that the influence of the effort or time for the actual allocation of tasks will be particularly evident when the workload of the agent is already high.

The task allocation decision is also bounded by a certain inhibitory bound or allocation preference threshold (Moray et al., 2000). This threshold determines when relative trust does not result in a preference difference high enough to rely on an agent.

Theory development on these factors is immature, but it is expected that the height of the threshold will be influenced by the difference between the trust uncertainty and the urgency and importance of the task allocation.

Finally, the task allocation decision is distinguished from the goal-directed actions of allocating a task or actually relying on an agent. The term *task allocation* is used to refer to the overt behaviors of agent i that are required to actually rely on agent j. The decision to rely on agent j may not be sufficient to reach the state in which the task is actually allocated to agent j. There may be unanticipated obstacles interfacing i and j that hinder the actual allocation of a task. This refers to the ability of the agent and opportunity in the environment. Furthermore, there can also be an action to allocate a task to an agent without a decision to allocate this task. This can be the case for instance when execution errors are made.

3 Formal Cognitive Model

Suppose a decision maker is given a (meta) task τ_m for which it has to make a best choice in allocating a certain (object) task τ_o to either a human agent H or a machine agent M. The *Decision Field Theory* (DFT) is a mathematical framework for describing the dynamics of such choices (Busemeyer and Townsend, 1992). In this section a formal model of task allocation decisions inspired on DFT is shown, which is used in describing the dynamics of the proposed experiment in Section 4.

The following formal model is described by means of four definitions, that is, of the *task execution state*, *trust state*, *allocation preference state*, and *preferred task execution state*. These are called states because they are time-dependent. The (preferred) task execution states are strings (sequences of characters). The trust and allocation preference states are real values.

Definition 1 (task execution state)**.** *Let σ_i be a* task execution state*:*

$$\sigma_i(j, \tau_o, t_n) = APPEND^n_{k=0} s_i(j, \tau_o, t_k) \quad (1)$$

where $i, j \in Agents = \{H, M, *\}$, $\tau_o \in Tasks$ and s_i is a recall function where $s_i : Agents \times Tasks \times Time \rightarrow Actions$, according to agent i. Agent $*$ represents the infallible agent. The function σ_i thus returns a string of sequentially ordered actions resulting from the execution of task τ_o by agent j according to agent i until time point t_n. Note that $\sigma_i(*, \tau_o, t)$ thus indicates the task execution state of the infallible agent according to agent i*. The function $APPEND$ appends an action at the tail of a given string.*

Example 1. *An example of a task execution state $\sigma_H(H, \tau_o, t_3) =$ "$\alpha_1\alpha_3\alpha_2\alpha_4$", where $\alpha_1, \alpha_3, \alpha_2, \alpha_4 \in Actions$ are the executed tasks at time points t_0, t_1, t_2, and t_3, respectively, and $H \in Agents$.*

The recall function s_i might result in actions falsely identified by agent i as executed on a certain time point by a certain agent. Such errors can be modeled by means of decays, e.g., by using a time-dependent randomization function. This means that $\sigma_i(j, \tau, t_n)$ is not necessarily the first part of $\sigma_i(j, \tau, t_m)$ for $t_n \leq t_m$ and arbitrary j (including $j = *$) and τ. In contrast, for $i = *$ the latter is not the case, which in other words means that the infallible agent has no regrets.

Similar to (Jonker and Treur, 1998), trust is considered a mental agent concept that depends on the past experiences that coincide on discrete time points with events that affect the agent's trust state. In this paper experiences are given by evaluating task execution states of an agent by means of comparison with those of the supposed infallible agent. This idea of the infallible agent and the comparison may be different for each agent.

Definition 2 (trust state)**.** *Let T_i be a* trust state*:*

$$T_i(j, \tau_o, t) = 1 - \frac{D_i(\sigma_i(j, \tau_o, t), \sigma_i(*, \tau_o, t))}{\mid \sigma_i(*, \tau_o, t) \mid} \quad (2)$$

where D_i is a function calculating the distance between two strings according to agent i. Trust states based on execution states with length 0, i.e., when $\mid \sigma_i(*, \tau_o, t) \mid = 0$, have initial values. Furthermore, $D_i(\sigma_i(j, \tau_o, t), \sigma_i(*, \tau_o, t))$ can also be written as the error rate $e_i(j, \tau_o, t)$.

The distance function D_i can be a *form* of the Hamming Distance (HD), i.e., for trust calculation

based on real performance history by means of 1-to-1 distance, or for instance the Levenstein Distance (LD), i.e., for determining model validity by means of the calculation of basic edit distance. The remaining of D_i is determined by agent i's interpretation and causal attribution resulting in inflation of penalties on errors due to for instance the workload and resource boundedness of agent j, complexity of τ_o, and memory decay, at time points $t_k \leq t$, or even $t_k > t$ when future events are anticipated in these terms. Three cases of memory decay are for instance modeled in (Jonker and Treur, 1998). Initial values of trust states, when $\mid \sigma_i(*, \tau_o, t) \mid = 0$, are determined by only such indirect indicators. Furthermore, all agents but $*$ can make errors or are biased in distance calculation, as in mistaken memory recalls and prejudices, respectively.

Example 2. *Please recall Example 1 of agent H. Let $\sigma_H(*, \tau_o, t_3) =$ "$\alpha_1\alpha_2\alpha_3\alpha_4$". Let's assume that exactly $D_{H,1} = HD$ is used. This means that trust state $T_H(H, \tau_o, t_3) = 1 - \frac{2}{4} = \frac{1}{2}$. But if we assume that exactly $D_{H,2} = LD$ is used, then the trust state $T_H(H, \tau_o, t_3) = 1 - \frac{1}{4} = \frac{3}{4}$. In this case always holds that $D_{H,2} \leq D_{H,1}$.*

Task allocation decisions are based on allocation preferences. As is proposed in (Gao and Lee, 2004; De Vries et al., 2003) the following model assumes that preferences are determined by trust in the self, trust in the other, and a certain corresponding inhibitory bound or allocation preference threshold.

Definition 3 (allocation preference state)**.** *Let P_i be an* allocation preference state*:*

$$P_i(\tau_o, t) = T_i(j, \tau_o, t) - T_i(i, \tau_o, t) \qquad (3)$$

where the trust state $T_i(j, \tau_o, t)$ means that agent i trusts agent j with respect to its performance in executing task τ_o at time point t. Agent i prefers allocation of τ_o to j iff $1 \geq P_i(\tau_o, t) > \theta_i(\tau_o, t)$ and to i iff $-1 \leq P_i(\tau_o, t) < -\theta_i(\tau_o, t)$ at time point t. The function θ_i represents the inhibitory bound of agent i. In other words, positive values for P_i indicate the tendency to allocate to the other and negative values to itself, if it exceeds a certain threshold $(-)\theta_i$. The real interval $[-\theta_i, \theta_i]$ indicates indifference of the agent i with respect to its allocation preference. The value of $\theta_i(\tau_o, t)$ depends on the characteristics of its parameters, such as decay due to costs of waiting (Busemeyer and Rapoport, 1988).

Example 3. *Please recall Example 2 of agent H. Suppose that $D_H = HD$, that $\sigma_H(M, \tau_o, t_3) =$ "$\alpha_2\alpha_2\alpha_3\alpha_4$", and thus $T_H(M, \tau_o, t_3) = \frac{3}{4}$, for another agent $M \in Agents$. This means that the allocation preference state $P_H(\tau_o, t_3) = \frac{3}{4} - \frac{1}{2} = \frac{1}{4}$. Hence, if $\theta_H(\tau_o, t_3) < \frac{1}{4}$, then at time point t_3 agent H prefers the allocation of task τ_o to agent M.*

The above does not yet take into account that task allocation decisions also concern the effort or time needed for engaging (re)allocation and all other consequences afterwards, such as task switching costs relating other tasks and additional overhead (like in Hoc, 2001). In fact, this may result in the opposite of what one might expect from mere difference in trust states. This thus suggests a different view of relative trust, namely trust relating the differences in desirability of the resulting outcome of *commencing* the allocation of a certain task to a certain agent, with respect to the overall system performance. In the context of the experiment proposed in the next section initially the first definition is chosen.

The allocation task τ_m itself can result in a task execution state $\sigma_i(j, \tau_m, t)$, trust state $T_i(j, \tau_m, t)$, and allocation preference state $P_i(\tau_m, t)$ with its inhibitory bound $\theta_i(\tau_m, t)$ for $i, j \in Agents$ by means of Equations 1, 2, and 3, respectively. In other words, this enables a decision maker to make preferred decisions on the allocation of the allocation task.

Definition 4 (preferred task execution state)**.** *Let π_i be a* preferred task execution state*:*

$$\pi_i(\tau_o, t_n) = APPEND_{k=0}^{n} s_i(j, \tau_o, t_k) \qquad (4)$$

where each agent $j \in Agents \setminus \{*\}$ is preferred at time point t_k by the preferred allocator determined by $\pi(\tau_m, t_n)$ according to agent $i \in Agents$.

Example 4. *Please recall Example 3 of agent H. Suppose that task τ_m is allocated to agent H. In this case the preferred task execution state $\pi_H(\tau_o, t_3) =$ "$\alpha_1\alpha_3\alpha_3\alpha_4$", because of allocation preference states indicating the preferred allocation of task τ_o to agent H, H, M, and M, at time points*

t_0, t_1, t_2, and t_3, respectively. This might be different if task τ_m is allocated to agent M at a certain time point, possibly due to differences in states, inhibitory bounds, recall, and distance functions.

Finally, true states are subscripted with a $*$, i.e., states according to the infallible agent; e.g., $\pi_*(\tau_o, t)$ denotes the actual preferred task execution state. Performance of a cooperative MAS is therefore calculated by means of $HD(\pi_*(\tau_o, t),\ \sigma_*(*, \tau_o, t))$.

4 Experiment Design

In order to validate implications of the theory introduced in Section 2 a simple experimental task is developed. The goal of this experimental task is to predict, as a human-machine team, the location of a disturbance. In every trial the disturbance can occur at one of three locations. Also each trial consists of three phases: a prediction phase, a selection phase, and an update phase. The human and the machine are both required to execute three tasks ($\tau_{o,m,u}$), one for each of these phases. The first task is to decide on the location of the next disturbance based on an internal prediction model. This decision is retrieved by letting both indicate a specific button. Given both predictions, the next task is to let them decide on which advise to trust the most based on their internal selection model.[1] This is again retrieved by letting both indicate a specific button, either following the prediction of itself, the other, both, or nobody. In the last phase the location of the disturbance is revealed according to a predetermined string $\sigma_*(*, \tau_o, t)$, which both agents are required to process by means of updating their internal models for task τ_o and τ_m. In Figure 1 the interface of a first implementation of the experimental environment is shown.

The independent variables are the error rates of the machine for each task, and the difficulty of the string. The error rate of the machine $e_*(M, \tau, t)$ is manipulated by having it choose $e_*(M, \tau, t) \cdot \mid \sigma_*(M, \tau, t) \mid$ times a random action in stead of the action $s_M(M, \tau, t)$, for each task τ and time point t. The difficulty of the string is manipulated by changing its length and generation rules, which has been subject in the study of human sequential processing some decades ago (e.g., Jones, 1971).

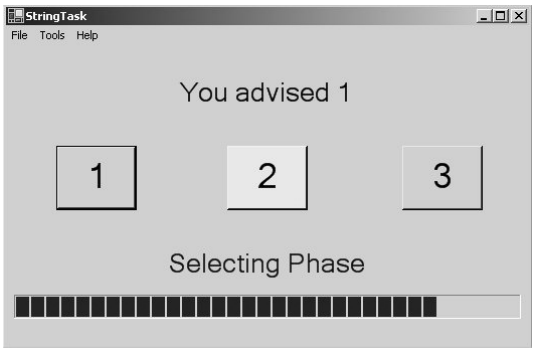


Figure 1. The interface of a first implementation of the experimental environment StringTask. A selection phase is shown, where the human predicted location 1 and the machine location 2. The allocator should indicate which button to select, based on both predictions and its internal selection model. After this the update phase indicates its soundness, which is used for updating the internal models.

The measured dependent variables are human-machine system performance and the error rates of the human for each task. These are simply calculated by means of the HDs of the preferred task execution state $\pi(\tau, t)$ and task execution state $\sigma_*(H, \tau, t)$, respectively, with the infallible task execution state $\sigma_*(*, \tau, t)$, for each task τ and time point t.

In the following experiment the effort and time to engage (re)allocation is kept the same for both human and machine. In order to ascertain that the experimental task can be reliably used to validate implications of the theory two straightforward hypotheses should hold:

- At each moment the participant prefers allocation of a task to the machine instead of to himself (or herself) when his trust in his own performance is expected to be significantly lower compared to his trust in the performance of the machine.
- At each moment the participant prefers allocation of a task to himself instead of to the machine when his trust in the performance of the machine is expected to be significantly lower compared to his trust in his own performance.

To validate the first hypothesis, the trust state

[1] This task is actually not a task allocation decision task in the precise sense of the definition given in Section 2. It is meant to catch an important prerequisite for the allocation decision, namely reasoning with allocation preference states.

$T_H(H, \tau_o, t)$ is experimentally manipulated by varying the error rate $e_H(H, \tau_o, t)$. This is done by decreasing the complexity of the string. If error rate $e_*(M, \tau_o, t)$ remains low enough, this ought to result in an allocation of the task τ_o to agent M by agent H, due to $1 \geq P_H(\tau_o, t) > \theta_H(\tau_o, t)$. In this experiment the task can be executed in three levels of difficulty. The level of difficulty is manipulated by increasing the memory-load of the internal prediction model that the agent H needs to use for executing task τ_o. It is known that human working memory has a limited capacity and that performance errors will result when more capacity is demanded by the task than can be supplied by the human. The memory-load of the internal models is manipulated by increasing the difficulty of the string.

Validation of the second hypothesis is symmetric. Trust in machine performance is manipulated by varying machine reliability. In this experiment agent M will perform the task at a reliability of 100, 70 and 50% independently of the difficulty of the task for agent H. In prior research it is often found that reliability lower than 70% will result in disuse of automation (Moray et al., 2000). The above manipulations result in a 3 (difficulty) $\times$ 3 (reliability) experiment design as shown in Table I.

It is expected that higher θ_H values will result in higher error rates $e_*(H, \tau_m, t)$ in the selection task due to unwanted indifference. Undoubtedly decision support is needed when in this diagonal region. How to support this and other results of this experiment will be subject of further experimental research.

5 Discussion

In this paper a computational model of trust based task allocation decision making and an experiment design used for theory validation are proposed. Though task allocation decision support by means of cognitive modeling of trust is clearly relevant, it is a field in AI that is quite new.

The present research is work in progress. After being confident on the replicability of previously found experimental findings in various domains in literature (Lee and Moray, 1992; Moray et al., 2000; Dzindolet et al., 2000; De Vries et al., 2003) by means of validating the two above mentioned hypotheses, the experimental environment will be used for further research, such as on indirect acquisition of knowledge (e.g., reputation, gossip), analogical judgments, allocation engagement costs (e.g., waiting, cooperation, and overhead costs), allocation implementation errors, level of autonomy, the allocation decision inhibitory bound, quantity and seriality of tasks, and time pressure. Extensions of (agent-based) cognitive models of trust and invocation concepts for machine monitoring of the allocation task (adaptive systems) are subject of investigation in the near future. Future research on cognitive modeling of trust aims at support in the four stages of information processing deliberation (Parasuraman et al., 2000): the acquisition of information relevant for trust, its integration to trust concepts, task allocation decision making based on trust concepts, and the implementation of the allocation decision. Moreover, future research foci on investigating the degree to which new or extended cognitive theories, based on formal modeling and controlled laboratory experiments, are translatable to more complex real world situations.

Acknowledgments

This research was done for the Human-Machine Task Integration Programme funded by the Dutch Ministry of Defense under programme nr. V206. Gratitude goes to Jan Maarten Schraagen, Egon van den Broek, and Jan Treur for their helpful comments.

References

Bliss, J., Dunn, M., and Fuller, B. S. (1995). Reversal of the cry-wolf effect - an investigation of two methods to increase alarm response rates. *Perceptual and Motor Skills*, 80:1231–1242.

Buehler, R., Griffin, D., and Ross, M. (1994). Exploring the "planning falacy": Why people underestimate their task completion times. *Journal of Personality and Social Psychology*, 67:366–381.

Busemeyer, J. R. and Rapoport, A. (1988). Psychological models of deferred decision making. *Journal of Mathematical Psychology*, 32:91–143.

Busemeyer, J. R. and Townsend, J. T. (1992). Fundamental derivations from decision field theory. *Mathematical Social Sciences*, 23:255–282.

De Vries, P., Midden, C., and Bouwhuis, D. (2003). The effects of errors on system trust,

Table I
THE PROPOSED 3 (STRING DIFFICULTY) × 3 (MACHINE RELIABILITY) EXPERIMENT DESIGN WITH THE EXPECTED PROPERTIES OF CORRESPONDING ALLOCATION PREFERENCE STATE P_H.

SD × MR	SD1	SD2	SD3
100% MR	$-\theta_H \leq P_H \leq \theta_H$	$1 \geq P_H > \theta_H$	$1 \geq P_H > \theta_H$
70% MR	$-1 \leq P_H < -\theta_H$	$-\theta_H \leq P_H \leq \theta_H$	$1 \geq P_H > \theta_H$
50% MR	$-1 \leq P_H < -\theta_H$	$-1 \leq P_H < -\theta_H$	$-\theta_H \leq P_H \leq \theta_H$

self-confidence, and the allocation of control in route planning. *International Journal of Human-Computer Studies*, 58:719–735.

Dijkstra, J. J., Liebrand, W. B. G., and Timminga, E. (1998). Persuasiveness of expert systems. *Behaviour and Information Technology*, 17:155–163.

Dzindolet, M. T., Beck, H. P., and Pierce, L. G. (2000). Encouraging human operators to appropriately rely on automated decision aids. In *Proceedings of the 2000 Command and Control Research and Technology Symposium*, Monterey, CA.

Dzindolet, M. T., Peterson, S. A., Pomransky, R. A., Pierce, L. G., and Beck, H. P. (2003). The role of trust in automation reliance. *International Journal of Human Computer Studies*, 58(6):697–718.

Dzindolet, M. T., Pierce, L. G., Beck, H. P., and Dawe, L. A. (2002). The perceived utility of human and automated aids in a visual detection task. *Human Factors*, 44:79–94.

Falcone, R. and Castelfranchi, C. (2001). Social trust: a cognitive approach. *Trust and deception in virtual societies*, pages 55–90.

Falcone, R. and Castelfranchi, C. (2004). Trust dynamics: How trust is influenced by direct experiences and by trust itself. In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04)*, pages 740–747, New York, USA.

Fuld, R. B. (1993). The fiction of function allocation. *Ergonomics in Design*, 1:20–24.

Gambetta, D. (1990). *Trust*. Basil Blackwell, Oxford.

Gao, J. and Lee, J. D. (2004). Information sharing, trust and reliance – a dynamic model of multioperator multiautomation interaction. In Vincenzi, D. A., Mouloua, M., and Hancock, P. A., editors, *Proceedings of the Second Human Performance, Situation Awareness and Automation Conference (HPSAA II)*, Daytona Beach, FL.

Hoc, J.-M. (2001). Towards a cognitive approach to human-machine cooperation in dynamic situations. *International Journal of Human-Computer Studies*, 54(4):509–540.

Jones, M. R. (1971). From probability learning to sequential processing: A critical review. *Psychological Bulletin*, 76(3):153–185.

Jonker, C. M. and Treur, J. (1998). Formal analysis of models for the dynamics of trust based on experiences. In Garijo, F. J. and Boman, M., editors, *Multi-Agent System Engineering, Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW'99*, volume 1647, pages 221–232, Berlin. Springer Verlag.

Kirlik, A. (1993). Modeling strategic behavior in human-automation interaction: Why an "aid" can (and should) go unused. *Human Factors*, 35:221–242.

Klein, G. (1998). *Sources of power: How people make decisions*. MIT Press, Cambridge.

Lee, J. and Moray, N. (1992). Trust, control strategies, and allocation of function in human-machine systems. *Ergonomics*, 35:1243–1270.

Lee, J. and Moray, N. (1994). Trust, self-confidence, and operators' adaption to automation. *International Journal of Human-Computer Studies*, 40:153–184.

Lee, J. D. and See, K. A. (2004). Trust in automation: Designing for appropriate reliance. *Human Factors*, 46(1):50–80.

Miller, C. A. (2002). Definitions and dimensions of etiquette. Technical Report FS-02-02, American Association for Artificial Intelligence, Menlo Park, CA.

Moray, N., Inagaki, T., and Itoh, M. (2000). Adap-

tive automation, trust, and self-confidence in fault management of time-critical tasks. *Journal of Experimental Psychology: Applied*, 6(1):44–58.

Parasuraman, R. and Riley, V. A. (1997). Humans and automation: Use, misuse, disuse, abuse. *Human Factors*, 39:230–253.

Parasuraman, R., Sheridan, T. B., and Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics*, 30:286–297.

Scallen, S. F. and Hancock, P. A. (2001). Implementing adaptive function allocation. *International Journal of Aviation Psychology*, 11:197–221.

Chapter 4

Closed-Loop Adaptive Decision Support Based on Automated Trust Assessment

This chapter appeared as (van Maanen et al., 2007b).

Closed-Loop Adaptive Decision Support Based on Automated Trust Assessment

Peter-Paul van Maanen[*†], Tomas Klos[‡] and Kees van Dongen[*]

[*] Department Human in Command, TNO Human Factors
P.O. Box 23, 3769 ZG Soesterberg, The Netherlands
Email: {peter-paul.vanmaanen, kees.vandongen}@tno.nl
[†] Department of Artificial Intelligence, Vrije Universiteit Amsterdam
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
[‡] Dutch National Research Institute for Mathematics and Computer Science (CWI)
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
Email: tomas.klos@cwi.nl

***Abstract*—This paper argues that it is important to study issues concerning trust and reliance when developing systems that are intended to augment cognition. Operators often under-rely on the help of a support system that provides advice or that performs certain cognitive tasks autonomously. The decision to rely on support seems to be largely determined by the notion of relative trust. However, this decision to rely on support is not always appropriate, especially when support systems are not perfectly reliable. Because the operator's reliability estimations are typically imperfectly aligned or calibrated with the support system's true capabilities, we propose that the aid makes an estimation of the extent of this calibration (under different circumstances) and intervenes accordingly. This system is intended to improve overall performance of the operator-support system as a whole. The possibilities in terms of application of these ideas are explored and an implementation of this concept in an abstract task environment has been used as a case study.**

1 Introduction

One of the main challenges of the Augmented Cognition Community is to explore and identify the limitations of human cognitive capabilities and try to let technology seamlessly adapt to them. This paper focuses on augmenting human cognitive capabilities concerning reliance decision making.

Operators often under-rely on the help of a support system that provides advice or that performs certain cognitive tasks autonomously. The decision to rely on support seems to be largely determined by the notion of relative trust. It is commonly believed that when trust in the support system is higher than trust in own performance, operators tend to rely on the system. However, this decision to rely on help is not always appropriate, especially when support systems are not perfectly reliable. One problem is that the reliability of support systems is often underestimated, increasing the probability that support is rejected. Because the operator's reliability estimations are typically imperfectly aligned or calibrated with true capabilities, we propose that the aid makes an estimation of the extent of this calibration (under different circumstances) and intervenes accordingly. In other words, we study a system that assesses whether human decisions to rely on support are made appropriately. This system is intended to improve overall performance of the operator-support system as a whole.

We study a system in which there is an operator charged with making decisions, while being supported by an automated decision support system. As mentioned above, the aim is to make the operator-support system as a whole operate as effectively as possible. This is done by letting the system automatically assess its trust in the operator and in itself, and adapt or adjust aspects of the support based on this trust. This requires models of trust, including a way of updating trust based on interaction data, as well as a means for adapting the type of support.

In this study, trust is defined as the attitude that an agent will help achieve an individual's goals,

possibly the agent itself, in a situation characterized by uncertainty and vulnerability (Lee and See, 2004). Trust can refer to the advice of another agent or to one's own judgment. Trust, like the feelings and perceptions on which it is based, is a covert or psychological state that can be assessed through subjective ratings. To assess trust, some studies have used scales of trust (e.g., Lee and Moray, 1992) and some studies have used scales of perceived reliability (e.g., Wiegmann et al., 2001). The latter is used because no operator intervention is needed. We distinguish trust from the decision to depend on advice, the act of relying on advice, and the appropriateness of relying on advice (Klos and La Poutré, 2006; van Maanen and van Dongen, 2005).

As a first implementation of this closed-loop adaptive decision support system, the operator-system task described in (van Dongen and van Maanen, 2006) has been extended.[1] This architecture instantiation leads to an overview of the lessons learned and new insights for further development of adaptive systems based on automated trust assessment. The present paper discusses some key concepts for improving the development of systems that are intended to augment cognition. The focus is on improving reliance on support.

In Section 2 an overview is given of the theoretical background of reliance decision making support systems and its relevance to the Augmented Cognition Community. In Section 3 the conceptual design of a reliance decision making support system is given. In Section 4 an instantiation of this design is described and evaluated. We end with some conclusions and future research.

2 Theoretical Background

The goal of augmented cognition is to extend the performance of human-machine systems via development and usage of computational technologies. Adaptive automation may be used to augment cognition. Adaptive automation refers to a machine capable of dynamic reallocation of task responsibility between human and machine. Reallocation can be triggered by changes in task performance, task demands, or assessments of workload. The goal of adaptive automation is to make human-machine systems more resilient by dynamically engaging humans and machines in cognitive tasks. Engaging humans more in tasks may solve out-of-the-loop performance problems, such as problems with complacency, situation awareness, and skills-degradation. This may be useful in situations of underload. Engaging machines more in tasks may solve performance degradation when the demand for speed or attention exceeds the human ability. This may be useful in situations of overload.

It should be noted that the potential benefits of adaptive automation turn into risks when the system wrongly concludes that support is or is not needed, or when the timing or kind of support is wrong (Parasuraman et al., 1999). For the adaptive system there may be problems with the real-time acquisition of data about the subject's cognition, with determining whether actual or anticipated performance degradations are problematic, and with deciding whether, when, and in what way activities need to be reallocated between human and machine. When the adaptive system is not reliable we create rather than solve problems: unwanted interruptions and automation surprises may disrupt performance and may lead to frustration, distrust, and disuse of the adaptive system (Parasuraman and Riley, 1997). In this paper we focus on computational methods that can be used to adjust the degree in which the machine intervenes.

When machine decisions about task reallocation are not reliable under all conditions the human operator should somehow be involved. One way is to make the reasoning of adaptive automation observable and adjustable for the operator. Understanding the machine's reasoning would enable her to give the system more or less room for intervention. Another and more ambitious way to cope with unreliable adaptive automation is by having a machine adjust its level of support based on a real-time model of trust in human reliance decision making capabilities. In this case it is the machine which adjusts the level of support it provides. The idea is adjusting the level of support to a level that is sufficiently reliable for the user, that problems with frustration, distrust and disuse of the adaptive system are reduced.

[1] A description and analysis of this system will be published in another paper in preparation.

A rational decision maker accepts support of an adaptive system when this would increase the probability of goal achievement and reject this support when it would decrease goal achievement. We rely more on support when we believe that it is thought to be highly accurate or when we are not confident about our own performance. People seem to use a notion of relative trust to decide whether to seek or accept support (Moray et al., 2000; Dzindolet et al., 2003; van Dongen and van Maanen, 2006). We also rely more on support when the decision of the system to provide support corresponds to our own assessment. The performance of an adaptive support system has to be trusted more than our own performance as well as be appropriately timed. In making a decision to accept support, users are thought to take the reliability of past performance into account. This decision to accept support is not based on a perception of actual reliability, but on how this is perceived and interpreted. Unfortunately, research has shown that trust and perceptions of reliability may be imperfectly calibrated: the reliability of decision support is under-estimated (Wiegmann et al., 2001; van Dongen and van Maanen, 2006). This could lead to under-reliance on systems that provide adaptive support. In this paper we argue that, because of this human bias to under-rely on support, reliance decision support designs are needed that have the following properties:

- **Feedback** They should provide feedback about the reliability of past human and machine performance. This would allow humans to better calibrate their trust in their own performance and that of the machine, and support them to appropriately adjust the level of autonomy of adaptive support.
- **Reliance** They should generate a machine's decision whom to rely on. Humans could use this recommendation to make a better reliance decision. This decision could also be used by the machine itself to adjust its level of autonomy.
- **Meta-reliance** They should generate a machine's decision whom to rely on concerning reliance decisions. This decision could combine and integrate the best reliance decision making capabilities of both human and machine. This could also be used by the machine itself to adjust its level of autonomy.

In the following sections we show how the above three functions could be realized by a system that automatically assesses trust in real-time.

3 Conceptual Design of Reliance Decision Support

In this section the three properties mentioned above are described in more detail, in terms of three increasingly elaborate conceptual designs of reliance decision support. First we abstract away from possible application domains in order to come to a generic solution. The designs presented in this section are applicable if the following conditions are satisfied:

- The application involves a human-machine cooperative setting concerning a complex task, where it is not trivial to determine whether the machine or the human has better performance. In other words, in order to perform the task at hand, it is important to take both the human's and the machine's opinion into account.
- Both the human operator and the automated aid are able to generate solutions to the problems in the application at hand. In other words, both are in principle able to do the job and both solutions are substitutable, but not necessarily generated in a similar way and of the same quality.
- Some sort of feedback is available in order for both machine and human to be able to estimate their respective performances and generate trust accordingly. In other words, there is enough information for reliance decision making.

In many cases, if for a certain task the above conditions do not hold (e.g., the operator's solution to a problem is not directly comparable to the aid's solution, or no immediate feedback is available), then for important subtasks of the task they generally still hold.

One could say that for all automated support systems the aid supports the operator on a scale from a mere advice being asked by the user, to complete autonomous actions performed and initiated by the aid itself. More specifically, for reliance decision making support, this scale runs from receiving

advice about a reliance decision, to the reliance decision being made by the aid itself. In a human-machine cooperative setting, a *reliance decision* is made when either the aid or the operator decides to rely on either self or other. In the designs presented below the terms *human advice* and *machine advice* refer to the decision made for a specific task. The terms *human reliance* and *machine reliance* refer to the reliance decisions made by the human and the machine, respectively, i.e., the advice (task decision) by the agent relied upon. Finally, the term *machine meta-reliance* refers to the decision of the machine whether to rely on the human or the machine with respect to their reliance capabilities.

3.1 Feedback

Agreement or disagreement between human and machine concerning their advice can be used as a cue for the reliability of a decision. In case of agreement it is likely that (the decision based on) the corresponding advice is correct. In case of disagreement, on the other hand, at least one of the advices is incorrect. To decide which advice to rely on in this case, the operator has to have an accurate perception of her own and the aid's reliability in giving advice. The machine could provide feedback about these reliabilities, for instance by communicating past human and machine advice performance. This would allow humans to better calibrate their trust in their own performance and that of the machine, and support them to adjust the machine's level of autonomy. In Figure 1 the conceptual design of machine feedback is shown.

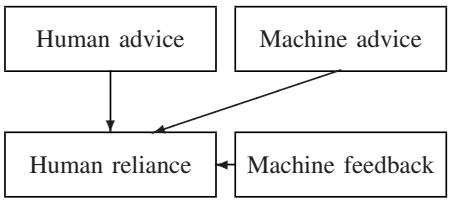


Figure 1. Both human and machine generate an advice on which the human's reliance decision is based. The machine provides feedback, for instance about the reliability of past human and machine performance. This allows humans to better calibrate their trust.

3.2 Reliance

Unfortunately, by comparing advice, one introduces an extra cognitive task: making a reliance decision. In this particular design the machine augments the cognitive function of reliance decision making, resulting in a decrease of the operator's workload. This can be in the form of a recommendation, or the reliance decision can be made autonomously by the machine, without any intervention by the human operator. The machine or human could adjust the machine's level of autonomy in that sense. Additionally, the human could provide feedback in order to improve the machine's decision. For instance, the human can monitor the machine in its reliance decision making process and possibly veto in certain unacceptable situations. In Figure 2 the conceptual design of such machine reliance is shown.

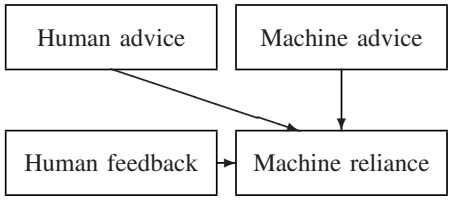


Figure 2. The machine generates a reliance decision. In this particular design the machine augments the cognitive function of reliance decision making. Both human and machine generate an advice on which the machine's reliance decision is based. It is possible that the human gives additional feedback.

3.3 Meta-reliance

Since in some situations humans make better reliance decisions, and in others machines do, reliance decision making completely done by the machine does not result in an optimal effect. Therefore, it may be desirable to let the machine decide whom to rely on concerning making reliance decisions. We called this process *meta-reliance decision making* and it combines the best reliance decision making capabilities of both human and machine. If the machine's meta-reliance decision determines that the machine itself should be relied upon, the machine would have a high level of autonomy, and otherwise a lower one. Hence the machine is capable of adapting its own autonomy. In Figure 3 the conceptual design of machine meta-reliance is shown.

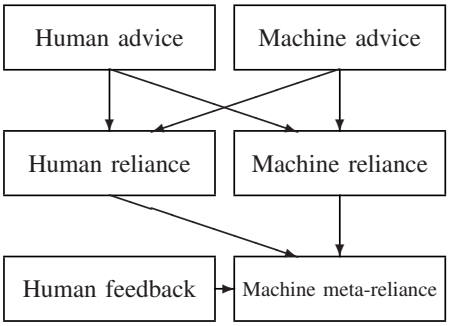


Figure 3. The machine generates a meta-reliance decision. It combines the best reliance decision making capabilities of both human and machine. Both the human and the machine generate advices and reliance decisions, on the latter of which the machine's meta-reliance decision is based.

4 Implementation and Evaluation

In this section we describe a proof-of-concept for the ideas presented above. In previous work (van Dongen and van Maanen, 2006), a collaborative operator-aid system was used in laboratory experiments to study human operators' reliance decision making. None of the additions described in Section 3 were employed, the setting was essentially that described in Section 3.1, without the aid's feedback. We have now extended the aid's design to provide the reliance and meta-reliance properties, and simulated the extended system's performance, compared to the results from the laboratory experiments. Below, we first describe the original and the extended task, and then the corresponding extensions in the aid's design. Finally, we present the improvements in system performance resulting from these additions.

4.1 The Task

For the experiment described in (van Dongen and van Maanen, 2006), participants read a story about a software company interested in evaluating the performance of their adaptive software before applying it to more complex tasks on naval ships. The story pointed out that the level of reliability between software and human performance was comparable and around 70%. Participants were asked to perform a pattern recognition task with advice of a decision aid and were instructed to maximize the number of correct answers by relying on their own predictions as well as the advice of the decision aid. The interface the participants were presented with is presented in the first 3 and the 6th rows of Figure 4. The task constitutes making a choice between 3 alternatives, as shown in each of the rows in the interface. In phase 1 the operator chooses, based on her own personal estimation of the pattern to be recognized. Then in phase 2 the machine chooses, with a pre-fixed average accuracy of 70%. Finally, in phase 3, the operator makes a reliance decision, by selecting the answer given in the first 2 phases by the agent she chooses to rely on. (The operator is free to choose a different answer altogether, but this happened only rarely in the experiments.) The last action of each trial consists of the feedback given by the system about which action was the correct one (phase 6), the corresponding button colored green if the operator's reliance decision was correct, and red if it was incorrect.

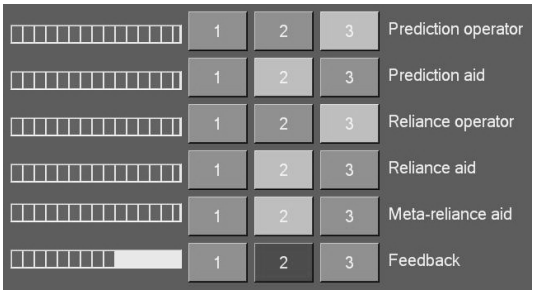


Figure 4. An example interaction between the operator and the automated decision aid. The rows represent the different phases of the operator-aid task. For the current research, phases 4 and 5 were added to the task environment described in (van Dongen and van Maanen, 2006).

In order to support the operator in making reliance decisions the above operator-aid task was extended by adding 2 phases representing the aid's reliance (Section 3.2) and meta-reliance (Section 3.3) decisions. The next section details the aid's design in this respect.

4.2 Design of the Aid

In the original experiments, the aid did nothing more than provide an advice to the human operator.

The enhancements to the aid's design were intended to provide the properties Reliance and Meta-reliance discussed in Section 3, to allow improvement upon the operator's Reliance Decision Making (RDM) in the form of Reliance Decision Making of the Machine (RDMM) and Meta-Reliance Decision Making of the Machine (Meta-RDMM).

Both RDMM and Meta-RDMM are based on a generic trust model (Klos and La Poutré, 2006) that allows the aid to estimate the operator's and the aid's abilities to make advice (task-related, prediction) and reliance decisions. The RDMM module makes the decision in phase 4 in Figure 4 ('Reliance Aid'), based on a comparison of the aid's trust in the operator's and the aid's own prediction abilities (phases 1 and 2). Like the operator in phase 3, the aid proposes in phase 4 the answer given in phases 1 and 2 by the agent it trusts most highly, where trust refers to *prediction* capability. In case of disagreeing reliance decisions in phases 3 and 4, the aid chooses among the operator and the aid in phase 5, this time based on a comparison of its trust in the two agents' *reliance decision making* capabilities.

As mentioned above, the same basic trust model is used for both estimates (prediction and reliance decision making capabilities). Essentially, the respective abilities are modeled as random variables $0 \leq \theta_a^x \leq 1$, which are interpreted as the probabilities of each of the agents $a \in \{\text{operator, aid}\}$ making the correct decision $x \in \{\text{prediction, reliance}\}$. The aid uses Beta probability density functions (pdfs) over each of these 4 random variables to model its belief in each of the values of $\theta \in [0, 1]$ being the correct one. Based on the feedback obtained in phase 6, each of the answers given in phases 1 through 4 can be classified as 'success' or 'failure' depending on whether the operator and the aid, respectively, were correct or incorrect in their prediction and reliance decisions, respectively. At the end of each trial, the aid uses Bayes' rule to update each of its estimates given the newly obtained information from phase 6. The advantage of using a Beta pdf as a prior in Bayesian inference about a binomial likelihood (such as that of θ), is that the resulting posterior distribution is again a Beta pdf (D'Agostini, 2003; Gelman et al., 2004).

In the next trial, the aid uses the new estimates about the agents' prediction abilities for RDMM in phase 4, and the estimates about the agents' reliance decision making abilities for Meta-RDMM in phase 5.

4.3 Experimental Results

The original experimental design and results are discussed in (van Dongen and van Maanen, 2006). Here, we show to what extent the elaborations of the aid's design were able to enhance the system's overall performance. Table I shows these results.

Table I
PERFORMANCE (PERCENTAGE CORRECT) OF OPERATOR RELIANCE DECISION MAKING (OPERATOR-RDM), RDMM, AND META-RDMM. PER ROW, THE DIFFERENCES BETWEEN OPERATOR-RDM AND RDMM, AND OPERATOR-RDM AND META-RDMM, ARE SIGNIFICANT.

	Operator-RDM	RDMM	Meta-RDMM
Exp. 1	0.65	0.70	0.70
Exp. 2	0.67	0.70	0.69
Both	0.66	0.70	0.69

Each participant played two experiments of 101 trials each. For each row, the improvements from operator reliance decision making (Operator-RDM) to RDMM, and from Operator-RDM to Meta-RDMM are significant. No significant difference in performance is found between RDMM and Meta-RDMM. There are no significant differences between experiment 1, 2, and both, for RDMM and Meta-RDMM. However, the differences between experiment 1, 2, and both, for Operator-RDM are significant. This means that, in our experiments, there was no measurable effect on performance of (Meta-)RDMM due to operator learning effects.

Our results indicate that the quality of the decision to rely on the prediction of either the operator or the aid is higher when it is made by RDMM than when it is made by human participants. When a computer would make reliance decisions based on RDMM it would outperform most human participants. However, it also became clear that in some situations humans make better reliance decisions than aids, and in others aids do. This means that reliance decision making completely done by the aid does not necessarily result in optimal performance. Meta-RDMM tries to take advantage of this and is based on the idea that the aid itself decides when to rely on RDMM and when to rely on the operator

for reliance decision making (meta-reliance). Our results show that Meta-RDMM also outperforms human participants in reliance decision making, but (surprisingly) significant differences between RDMM and Meta-RDMM were not found.

5 Conclusion

The goal of augmented cognition is to extend the performance of human-machine systems via development and use of computational technology. In the context of the current work, performance can be improved when, like in human-human teams, both human and machine are able to assess and reach agreement on who should be trusted more and who should be relied on in what situation.

In this paper we showed that human reliance decisions are not perfect and reliance decision making can be augmented by computational technology. Our machine reliance decision making model outperforms human reliance decision making.

Now that we have our proof-of-concept in an abstract task, we intend to investigate how human-machine cooperation can be augmented in more complex and more realistic situations. We intend to focus on how models of trust and reliance can be practically used to adjust the level of autonomy of adaptive systems. We want to investigate in what domains this kind of support has an impact on the effectiveness of task performance, and how the magnitude of the impact depends on the task's and the domain's characteristics. How serious are the conditions mentioned in section 3, both in terms of limiting the scope of application domains, and in terms of determining the effectiveness of our solutions. An important question is whether the properties of our abstract task environment are paralleled in real-world settings.

Acknowledgments

This research was partly funded by the Royal Netherlands Navy under program number V206 and by the Dutch government (SENTER) under project number TSIT2021.

References

D'Agostini, G. (2003). Bayesian inference in processing experimental data: Principles and basic applications. *Reports on Progress in Physics*, 66:1383–1419.

Dzindolet, M. T., Peterson, S. A., Pomransky, R. A., Pierce, L. G., and Beck, H. P. (2003). The role of trust in automation reliance. *International Journal of Human Computer Studies*, 58(6):697–718.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004). *Bayesian Data Analysis*. Chapman & Hall/CRC, second edition.

Klos, T. B. and La Poutré, H. (2006). A versatile approach to combining trust values for making binarydecisions. In *Trust Management*, volume 3986 of *Lecture Notes in Computer Science*, pages 206–220. Springer.

Lee, J. and Moray, N. (1992). Trust, control strategies, and allocation of function in human-machine systems. *Ergonomics*, 35:1243–1270.

Lee, J. D. and See, K. A. (2004). Trust in automation: Designing for appropriate reliance. *Human Factors*, 46(1):50–80.

Moray, N., Inagaki, T., and Itoh, M. (2000). Adaptive automation, trust, and self-confidence in fault management of time-critical tasks. *Journal of Experimental Psychology: Applied*, 6(1):44–58.

Parasuraman, R., Mouloua, M., and Hilburn, B. (1999). Adaptive aiding and adaptive task allocation enhance human-machine interaction. *Automation Technology and Human Performance: Current Research and Trends*, 22:119–123.

Parasuraman, R. and Riley, V. A. (1997). Humans and automation: Use, misuse, disuse, abuse. *Human Factors*, 39:230–253.

van Dongen, K. and van Maanen, P.-P. (2006). Under-reliance on the decision aid: A difference in calibration and attribution between self and aid. In *Proceedings of the Human Factors and Ergonomics Society's 50th Annual Meeting*.

van Maanen, P.-P. and van Dongen, K. (2005). Towards task allocation decision support by means of cognitive modeling of trust. In Castelfranchi, C., Barber, S., Sabater, J., and Singh, M., editors, *Proceedings of the Eighth International Workshop on Trust in Agent Societies (Trust 2005)*, pages 168–77.

Wiegmann, D. A., Rich, A., and Zhang, H. (2001). Automated diagnostic aids: the effects of aid reliability on user's trust and reliance. *Theoretical Issues in Ergonomics Science*, 2:352–367.

Chapter 5

Reliance on Advice of Decision Aids: Order of Advice and Causes of Under-Reliance

This chapter is partly based on (van Dongen and van Maanen, 2005, 2006b,a).

Reliance on Advice of Decision Aids: Order of Advice and Causes of Under-Reliance

Peter-Paul van Maanen*†, Kees van Dongen* and Lisette de Koning*

* TNO Human Factors, P.O. Box 23, 3769 ZG Soesterberg, The Netherlands
Email: {kees.vandongen, peter-paul.vanmaanen, lisette.dekoning}@tno.nl
† Department of Artificial Intelligence, Vrije Universiteit Amsterdam
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands

***Abstract*—This study investigates the effects of order and reliability of advice on reliance on decision aids. Effects are measured in terms of attribution of errors, estimation of reliability, understandability of processes, and accuracy motivation. Anchoring effects predict more reliance on advice when provided before the own judgment. Attribution biases predict underestimation of the reliability of the decision aid. Asymmetry in understandability of process predicts reliance on self. Accuracy motivation predicts willingness to accept advice. 79 Participants performed an uncertain pattern learning task with a decision aid and received performance feedback. Advice was presented before or after the own judgment. Participants chose to rely on their own judgments or on advice of the decision aid. Reliance on the decision aid was not higher when advice was presented before the own judgment, surprisingly perceived reliability of self was. Operators did not rely more often on the decision aid when in disagreement, although they perceived it to be 30% more reliable. Errors of the decision aid were less attributed to temporary and uncontrollable causes and its reliability was underestimated persistently. Reliance on self was not only predicted by a biased relative trust, but also by relative understandability and responsibility felt for accuracy. Cuing effects are found, but only when people trust themselves more than the decision aid. Under-reliance can be caused by asymmetries in estimation of reliabilities and attribution of errors, asymmetries in understandability of underlying process, and low accuracy motivation. These findings are potentially applicable for the design of decision aids and training procedures.**

1 Introduction

Information and communication technology is changing the nature of work. The use of decision aids in complex systems, such as aviation, nuclear power, health care or command and control is becoming increasingly common. The assumption behind the introduction of decision aids is that a team of human and decision aid will be more effective than either human or decision aid working alone. Performance improvement by introducing decision aids is difficult to predict, because decision aids are not always used appropriately. It is often found that users tend to rely too much or too little on decision aids (Parasuraman and Riley, 1997). For instance, Skitka et al. (1999) found that unaided participants made fewer errors than participants who worked with a decision aid. The last group relied too much on the aid and missed events that they could have discovered manually. Like human operators, in complex domains, it is not likely that decision aids are 100% reliable. A problem with decision aids is that these systems often have incomplete or unreliable data or knowledge and use simplifying assumptions that make them brittle (Guerlain et al., 1999). This means that users cannot blindly accept advice of a decision aid; sometimes they need to reject advice and rely on their own decision. The tendency to accept advice depends among others on the reliability of the decision aid.

The relationship between the reliability of the decision aid and the users' reliance on decision aids is complex and multifaceted (Thomas and Rantanen, 2006; Parasuraman and Riley, 1997; Dzindolet et al., 2003; Lee and See, 2004). It seems not possible to determine a fixed threshold for an acceptable level of unreliability (Thomas and Rantanen, 2006) (though Wickens and Dixon (2007) suggest a threshold level of 70%). Reliance on advice is mediated by a range of cognitive variables of which trust in oneself (self-confidence) and trust in the

decision aid are two central concepts. To increase the effectiveness of human-computer collaboration several frameworks have been developed to better understand how people use decision aids.

This paper is composed of the following sections. In Section 2 related work on the psychological effects on reliance on decision aids is discussed. In Section 3 our contribution is explained by posing several research questions and motivating a number of hypotheses with respect to 1) the self bias and the order of advice, 2) understandability of underlying reasoning, 3) feeling of responsibility, 4) accuracy of perceived reliability and 5) the attribution bias. After this, in Section 4 the method of the experiment is described of which the results are given in Section 5. In Section 6 the paper ends with a discussion and conclusions.

2 Background

Several frameworks of trust have been developed to identify factors that affect reliance on decision aids (e.g., Dzindolet et al., 2001; Lee and See, 2004). The framework of Dzindolet et al. (2001) emphasizes cognitive, social and motivational factors in reliance on decision aids. Concerning cognitive factors, users for instance compare the perceived reliability of the decision aid with how they perceive their own. Whether this leads to appropriate reliance on the advice of the decision aid, depends on whether these perceptions of reality correspond to reality. Dzindolet et al. (2001) however have suggested that users often wrongly expect that decision aids perform nearly perfect due to their seemingly infallible calculation capabilities. Another example of a cognitive factor is that users sometimes rely on decision aids to save mental effort. Mosier and Skitka (1996) have used the term 'automation bias' to refer this tendency.

Concerning social factors, trust is an important factor. People rely more on decision aids when they trust the decision aid more than themselves (Mosier and Skitka, 1996). Other examples of social factors are: feelings of control and moral obligation to rely on oneself. Finally, motivational factors such as the effort invested in relying on oneself or decision aid are also part of the framework. Motivation is affected by context factors such as workload and penalties or rewards when (not) using decision aids.

The strengths of the framework of Dzindolet et al. (2001) is that it identifies many psychological factors that may affect reliance decisions. A disadvantage of the framework is that it is less specific on the dynamics of trust and reliance and less specific on the appropriateness of trust.

The framework of Lee and See (2004) and that of Gao and Lee (2006) emphasize the dynamics of trust in, and reliance on, automation and take into account the role of feedback. Trust is not static, it changes in time as it is influenced by direct and indirect sources of knowledge. Trust is defined as the attitude that an agent will help achieve an individual's goals in a situation characterized by uncertainty and vulnerability. A distinction is made between, trust itself on the decision aid, the decision to rely on it, the act of reliance on it, and the appropriateness of this reliance on the decision aid (van Dongen and van Maanen, 2006). Trust is affected by positive and negative experiences with the decision aid; by reputation and gossip about it, but also by properties of the user such as the propensity to trust (Merritt and Ilgen, 2008). It can be based on analogical judgments, judgment about the trustworthiness of a category rather than on the actual performance of one of its presumed members. Etiquette may for instance influence trust because category membership associated with adherence to a particular etiquette helps people to infer how automation will perform (Miller, 2002). Like the framework of Dzindolet et al. (2001) this framework recognizes relative trust as a basic component of decisions about reliance: reliance is determined by the difference between a decision maker's trust in a decision aid and the confidence he has in his own performance. If this difference exceeds a particular threshold, i.e., when the trust in the decision aid is higher by some amount than the decision maker's self-confidence, then he will switch from relying on oneself to relying on the decision aid and vice versa. Trust, in turn, depends upon previous performance of oneself or decision aid. This creates a feedback loop which is an important element of the framework (Gao and Lee, 2006).

The framework of Lee and See (2004) also emphasizes how changes in trust and reliance are affected by system factors such as interface features; factors such as the tendency to trust; by

organizational factors such as gossip and reputation; cultural factors such as norms and expectations; and task and context factors such as workload and time constraints. In addition to a focus on the dynamics of trust this framework also provides concepts to determine the appropriateness of trust. It integrates concepts like trust calibration.

3 Hypotheses

In this section the research questions and hypotheses on which the present research is based are given and motivated. We try to give answers to the following research questions:

1a) When do we observe a self bias rather than an automation bias?
1b) What is the effect of relative trust and the order in which advice is presented (i.e., either before or after making one's own opinion explicit) on these biases?
2a) Can performance be used to accurately predict reliance behavior on a decision aid?
2b) What is the effect of asymmetric availability of to the reasoning underlying the advice of the decision aid and one's own decision making?
3) Does the level of responsibility felt for achieving the task outcome influence reliance on advice?
4) Does feedback about performance result in over- or underestimation of the reliability of oneself and the decision aid?
5) Are the causes of unreliability differently attributed for oneself and decision aid?

The above questions 1 (a and b), 2 (a and b), 3, 4 and 5 are further discussed in sections 3.1, 3.2, 3.3, 3.4 and 3.5, respectively. These discussions lead to 7 hypotheses.

3.1 Self Bias and Order of Advice

In many operational settings, it takes less mental effort to use advice of a decision aid than to vigilantly seek and process information oneself. This however is not always the case. In some situations it takes more effort to seek, accept and use advice of a decision aid than to rely on one's own opinion. For the same reason people heuristically rely on a decision aid, people may heuristically rely on their own thinking: that is to save mental effort. The *self bias* or *self heuristic* is expected to be observed when the effort to rely on one's own judgment is perceived to be lower than the effort required by accepting advice of a decision aid. This may for instance be the case when one's own judgment is cognitively more 'available' than advice of a decision aid or when advice of a decision aid conflicts with one's initially held beliefs. According to the *availability heuristic* people base their prediction on how easily knowledge can be brought to mind (Tversky and Kahneman, 1973). This suggests that the availability of one's own judgment relative to that of the decision aid will affect whether an *automation bias* or a self bias is induced.

The degree to which advice is available to one's own judgment may for instance be influenced by the order in which advice is provided by decision aids. Advice of a decision aid can either be presented before or after the decision maker has formed his own opinion. When advice is presented first, people can automatically follow that advice without thinking about the problem themselves. This causes their own knowledge about the decision problem at hand to be mentally less available. Sniezek and Buckley (1995) found that the answers of participants who received advice first, matched more often with the answers of the advisor compared to participants who first formed their own opinion. This tendency to rely on advice may be reduced by actively involving humans in decision making. This can be done by providing advice after the decision maker has formed his own judgment, instead of before. This is for instance also done in critiquing systems (Guerlain et al., 1999; Silverman, 1992). When advice is presented after people have formed their initial decision, they cannot automatically follow advice. They are required to first think for themselves, which makes their own cognitions more available.

However, presenting the advice after an own judgment is formed might cause other problems. The literature on decision making suggests that people are often reluctant to change their mind; tend to commit to their initial judgments; wish to be consistent in their thoughts and actions; tend to ignore or under-utilize conflicting information or simply tend to rely on the first alternative that is good enough.

Further, advice that is provided after one has

formed one's own judgment makes possible disagreement with a decision aid also more salient. Madhavan and Wiegmann (2007), for instance, found that when participants were forced to answer before they received advice of a decision aid, they disagreed more often with the decision aid compared to participants who first received advice.

Together with the availability effect, this tendency to stick to one's initially formed judgment may induce a self bias when advice is presented after one's own judgment. Changing the order of advice takes the human in the loop and reduces the automation bias, but may be replaced by a self bias.

The influence of order of advice on reliance behavior is also expected to be affected by relative trust. Relative trust is the difference between trust in own performance and trust in that of the decision aid. Relative trust is for an important part determined by perceptions of reliability (Lee and See, 2004; Gao and Lee, 2006). When people trust the decision aid more than themselves we think that the tendency to rely on oneself is not induced. This boils down to the following hypothesis:

Hypothesis 1. *The self bias is observed when conflicting advice is presented after people have formed their initial judgment, but only when they trust themselves more than the decision aid.*

3.2 Understandability of Underlying Reasoning

In contrast to one's own reasoning, the reasoning of a decision aid is often not easily accessible or understandable to the user, especially when the decision aid is a computer rather than a human. At best, only part of the decision aid's reasoning can be made transparent. However, this is often not understandable and therefore cognitively not available. In most cases, only a small part of the decision aid's reasoning is transparent. Yaniv and Kleinberger (2000) argue that advice is often underused because decision makers have direct access to the reasons supporting their own judgment as well as to the strength of those reasons, but often have no direct access to the reasons underlying the advice of an advisor or in this case a decision aid. A common assumption in cognitive psychology is that the weight placed on a judgment depends on the evidence that is recruited to support that judgment (Tversky and Koehler, 1994). Because the processes underlying the decision aid's advice are less available and understandable compared to one's own judgment, we expect that reliance on advice of the decision aid cannot be solely predicted by relative trust in performance reliability.

The above leads to the following two hypotheses:

Hypothesis 2. *Decision makers rely less on the decision aid than would be expected based on relative trust in performance reliability alone.*

Hypothesis 3. *Decision makers rely less on conflicting advice when they perceive the advisor's reasoning to be cognitively less available and understandable than their own reasoning.*

3.3 Feeling of Responsibility

Working in a team has advantages and disadvantages. Although two may know more than one, a disadvantage of teamwork may be that responsibility may diffuse between its members and that they do not feel accountable for the task outcome. Several researchers think of the humancomputer system as a team in which one member is not human (e.g., Bowers et al., 1996). The human may feel less responsible for the outcome when working with a decision aid than when working alone and may invest less mental effort. It is expected that the less responsible the person feels, the less motivated the person is to invest mental effort in the task and the more likely he is to act heuristically. In situations that induce the automation heuristic one expects users to heuristically rely on advice. In situations that induce a self bias, however, we expect the opposite effect. When disagreement with the decision aid is salient, we expect people to invest the mental effort that is needed to overcome commitments to initial judgments, but only when they feel responsible for the task outcome. When the felt responsibility for task outcome is low, we expect that people do not change their mind and tend to reject conflicting advice. This results in the following hypothesis:

Hypothesis 4. *In situations that induce a self bias, people who feel more responsible for the task outcome, rely more on conflicting advice than people who feel less responsible.*

3.4 Accuracy of Perceived Reliability

The above mentioned psychological concepts (i.e., perceived reliability, relative trust, cognitive availability, understandability and responsibility) may explain how reliance decisions are made, but do not explain whether reliance on advice is appropriate or not.

For appropriate reliance on advice one would expect a rational decision-maker to rely on advice when this would increase the probability of goal achievement and to reject advice when it would decrease this probability. The decision to accept or reject advice is, however, not based on a comparison of the actual reliability of oneself or decision aid, but on how these are perceived. Unfortunately these perceptions not necessarily correspond to reality and may be prone to random and systematic errors. Perceived reliability of oneself and decision aid may be under-estimated or over-estimated and when the direction or magnitude of error differs between oneself and decision aid this could lead to over-reliance or under-reliance on advice.

Concerning perception of one's own performance, studies of judgment under uncertainty have indicated that humans are often over-confident (e.g., Alba and Hutchinson, 2000). An explanation for this is that people tend to focus on supporting rather contradictory evidence for a judgment, decision or prediction. Although pervasive in the literature, over-estimation of one's own performance is not universal (Brenner et al., 1996). May (1987, 1988)'s results for instance yielded 9% over-confidence when confidence in performance was estimated after each answer, whereas a 9% under-confidence was found when confidence in performance was estimated after each block. An explanation for this is that estimated percentage correct is likely to be based on a general evaluation of the difficulty of the task or based on feedback about performance, rather than on a balance of arguments for and against each specific judgment (Brenner et al., 1996). Whether over- or underestimation of one's own performance is observed seems to depend on how and when people are asked to estimate their performance rate.

Concerning the perception of the decision aid's performance, Wiegmann et al. (2001) found that it is often underestimated. One reason for this may be that decision aids do not perform as expected. Dzindolet et al. (2001) argue that the perception of the reliability of an automated decision aid is filtered through the operator's 'perfect automation schema' or expectation that automation will perform at near perfect rates. This sometimes unrealistic expectation may lead operators to pay too much attention to information that is in conflict with the schema: errors. Consequently, errors made by automation trigger a rapid decline in trust when decision aids make errors (Dzindolet et al., 2002). Whether the decision aid's performance is over- or underestimated depends on what level of performance is expected in advance.

Providing users of decision aids with realistic information about the user's reliability and that of the decision aid results in more appropriately calibrated trust. Although performance feedback is expected to improve the accuracy of perceived reliability of oneself and decision aid, it is not expected to lead to a perfect correspondence. It has been argued that trust is a nonlinear function of performance and that it tends to be conditioned by negative experiences. Negative experiences have a greater influence on the perception of the reliability of the decision aid than positive experiences (Lee and See, 2004). Although performance feedback is expected to improve perceptions of reliability, underestimation of performance is expected because of this negativity effect. This leads to the following hypothesis:

Hypothesis 5. *Perceived reliability of both oneself and decision aid is underestimated when feedback about performance is provided.*

3.5 Attribution Bias

Reliance on decision aids is not only affected by beliefs about performance reliability itself, but also by beliefs about the processes that affect this performance (Lee and See, 2004). When the processes underlying the decision aid's advice and the factors that affect the reliability of these processes are not observable, causes of unreliability are *inferred* instead of observed. According to Weiner (1986)'s attribution theory, these causal attributions result in affective reactions, which may affect the level of trust in the decision aid or oneself. The attribution theory claims that how people assign success and

failure can be divided into three categories. The first is internal or external attribution (locus). External attribution means that performance is perceived to be influenced by attributes outside the decision aid, such as the dynamics, complexity or unpredictability of the task. Internal attribution means that performance is perceived to be influenced by factors inside the decision aid, such as the competence or motivation to perform the task. The second is attribution to factors that are temporary or permanent (stability). When errors are thought to be caused by temporary factors, more optimism is expected than when errors are attributed to permanent attributes of the agent or task. The third category is attribution to factors one can or cannot control (controllability). When errors are assigned to causes that are not under control (e.g., unpredictability of situation) people are more forgiving than when errors are perceived to be under control (e.g., motivation).

Unfortunately, people are known to be biased in how causes are attributed to success and failure and asymmetries in attribution of one's own performance to causes and that of others are often found. One common bias in assigning causes is called the *fundamental attribution error* or *correspondence bias*. This is the tendency of people to under-emphasize situational causes for the behavior of others. Our own errors are more likely to be attributed to temporary, external or uncontrollable factors, while errors of others are more likely to be attributed to permanent, internal and controllable factors. In other words, we have excuses for our own errors, but not for others. Gilbert and Malone (1995) point out that for a correct attributional analysis that takes into account the role of situational causes for the behavior of others one must not only have realistic expectations about their performance but also perceive and recognize situational constraints for the other. The problem is, however, that factors that constrain the reliability of decision aids, such as the unreliability of the data it uses or the inherent unpredictability of the situation it operates in is often not known or observable for the user. Because situational causes that constrain the users task performance are more salient from the user's perspective than those that constrain the performance of the decision aid these causes are also be expected to be cognitively less available when causal attributions are made. As a result users are expected to be less forgiving and less optimistic about the performance of the decision aid than about their own performance.

Since one can think of the human-computer system as a team in which one member is not human (e.g., Bowers et al., 1996), one can also think of the theory on causal attribution in humans alone, to also hold in the context of human-computer collaboration. This would lead to the following hypotheses:

Hypothesis 6. *Underestimation of the reliability of the decision aid is expected to be more prevalent and more persistent than underestimation of reliability of oneself.*

Hypothesis 7. *Unreliability of the decision aid is less attributed to temporary, external and uncontrollable causes.*

4 Method

4.1 Participants

79 College students participated in the experiment. Ages ranged from 18 to 38 years ($M = 23$). Participants were paid € 35 for their participation.

4.2 Apparatus

4.2.1 Task and Procedures: Before the training and experimental trials participants read a cover story about a software company interested in evaluating the performance of their pattern learning software before applying it to more complex tasks on naval ships. To neutralize the effect of unrealistic expectations (i.e., perfect automation schema) the story pointed out that the level of reliability of both software and human performance was imperfect and, depending on the amount of training, was correct for 70% of the time. This level was chosen because by this threshold humans tend to flip between relying on themselves and a decision aid (Wickens and Dixon, 2007). Prior pilots also showed that this was indeed the case.

Participants were asked to maximize the number of correct final predictions by relying on their own predictions as well as the advice of the decision aid. The task required participants to predict what number (1, 2 or 3) would occur in the present trial

(see for instance row one in Figure 1). This prediction had to be based on the gradual discovery of a repeated pattern of numbers revealed in previous trials. The pattern used (i.e., 2, 3, 1, 2, 3) was repeated until a sequence of 100 numbers was formed. These numbers were then partly randomized (10%). This randomization was done in order to control the difficulty of detecting the pattern and make the participants think they still did not fully find the correct pattern (and otherwise performance would become 100% after a while; for more on this see Section 4.2.3). In each trial the correct number was revealed by the highlighted button on the last row of the interface (see Figures 1 and 2).

After the instructions participants performed 40 practice trials in which they had to discover a pattern in the data. The sequence of numbers for the practice trials was constructed in a similar way as described above. The participants could experience that their own performance and the advice of the decision aid was not perfect. For the first (practice) trials participants had no information about the correct sequence of numbers and could only guess. After a few trials, participants could form a more or less stable, but imperfect, mental model of the pattern of numbers based on the feedback they received. By building up, remembering, using and adjusting this model, participants could predict with some degree of success (i.e., aiming at an on average success of around 70% under normal conditions) which button should be pushed next. After the training trials the actual experiment started.

To be able to observe possible learning effects each participant performed two experimental blocks, each consisting of 100 trials. After each experimental block participants had to fill in questionnaires. Between each two blocks participants had a short break.

4.2.2 Reliability of Decision Aid: The actual reliability of the decision aid was set to vary between 60 and 80% with an average reliability of 70% (similar as the human performance) and an SD of 3% for each block (100 trials). Errors were defined as a deviation from the correct pattern of answers as provided by the feedback (last row). The average of 70% was used since the success of this experiment depended on an on average equal amount of situations where participants could rely on themselves or on the decision aid (i.e., an average reliability of, for instance, 90% or 50% would result in less challenging reliance decisions for the participants; A similar argument can be given for the choice for aiming for a performance average of 70% as suggested by Wickens and Dixon (2007)). The causes of unreliability were not made transparent to the user and were made to occur at random intervals such that the time of occurrence could not be anticipated.

4.2.3 Task Predictability: Since the reliability of individual participants was not under experimental control, we controlled the predictability of the task which influences the reliability of their advice. Ten percent of the pattern in the sequence of numbers (last row) differed randomly from what would be expected by extrapolating the dominant and recurring pattern (i.e., ..., 2, 3, 1, 2, 3, ...). Like the unreliability of the decision aid's advice, the unpredictability of the pattern occurred at random intervals. This made the decision to rely on oneself or the decision aid more difficult. Without the control of task predictability, floor or ceiling effects in the performance of reliance could occur, i.e., when the reliability of the participants' advice becomes predictable, their reliance decision also becomes predictable. The used sequence of numbers (of length 100) was determined beforehand and tested to have a Hamming distance of 10. Pilots (and later post-experimental questionnaires) showed that this partial randomization was enough to vary in a controlled manner the difficulty of the pattern. Participants did not suspect any randomization and did not find out that the pattern would never be discovered. Participants just suspected that their idea of what the pattern should be was imperfect and hence the confidence in oneself decreased (or increased when they were right). If the decision aid was correct, the confidence in the decision aid increased.[1] This was due to the fact that humans tended to see patterns in noise and because of the convincing story told in the beginning of the experiment (which was also both confirmed in post-

[1]Note that the reliability of the decision aid was not affected by randomization for the control of task predictability. Randomization for adapting the reliability of the decision aid was done using the already randomized sequence which was used to give feedback to the participant.

experimental questionnaires). Appropriate calibration to these occurrences would lead to appropriate improved reliance decisions. Of course participants were unable to predict whether there was an instance of randomization, but the mentioned reliance decision could be made independent of the advices given (i.e., the reliance decision could already be made before the advices of oneself and the decision aid are given).

4.3 Design

A between-subjects design was used with 'order of advice' as independent variable. Participants were randomly allocated to either the 'human first' (40) or 'decision aid first' (39) condition.

4.4 Independent Variables

Order of Advice: In the 'human first' condition the order of activities in each trial was: predict; receive advice from decision aid; revise prediction (or re-select same prediction); and receive feedback with the correct answer, which corresponds to each row in Figure 1, respectively. Participants first made their own independent prediction (initial decision) by clicking on one of the three numbers in the first row. Then the decision aid communicated its advice by highlighting one of the three numbers on the second row. Neither the data nor the rules, on which this advice was based, were made transparent to the participant. On the third row participants had to formulate their answer again (final decision) and were allowed to revise their initial decision. When their initial prediction differed from the advice of the decision aid, they could either follow their own initial prediction or the advice of the decision aid. On the fourth row the correct answer (highlighted button) and feedback about the success of the final decision (red or green color) was provided. By comparing the correct answer with the responses on the first three rows participants were able to calibrate their perceptions of the reliability of 1) their own initial predictions, 2) the decision aid's advice, and 3) the reliability of their final decisions. This calibration is expected to determine the decision to rely on the advice of themselves or on that of the decision aid.

In the 'decision aid first' condition the order of activities in each trial is: receive advice from decision aid; predict and receive feedback with correct answer, which corresponds to each row in Figure 2, respectively. In each trial participants first received advice before they could express their own prediction. On the second row the participants expressed their own prediction. They could follow the advice of the decision aid or make their own prediction that. On the third row, the correct answer (highlighted button) and feedback about the success of the decision (red or green color) was provided.

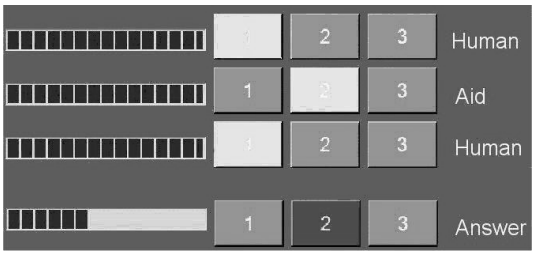


Figure 1. Interface of the pattern learning task: Human first.

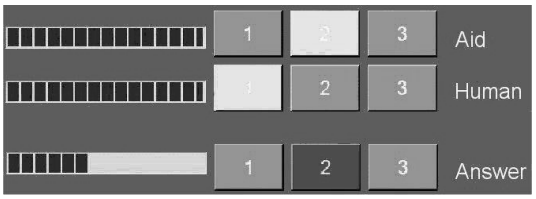


Figure 2. Interface of the pattern learning task: Decision aid first.

4.5 Dependent Variables

Agreement with the Decision Aid: Percentage agreement or matching with the decision aid is measured during experimental trials and is defined by the correspondence of the final decision of the participant with the advice of the decision aid (Bonaccio and Dalal, 2006). Percentage agreement allows us to compare the degree to which participants rely on themselves rather than on the decision aid in both conditions.

Agreement or matching measures are insensitive to changes in pre-advice and post-advice decisions. They cannot distinguish between whether one agrees with a decision aid because one is holding to one's pre-advice decision or whether one adopts

advice of a decision aid that conflicts with one's pre-advice decision. However, only the latter is a true decision to rely on the decision aid. To determine what factors affect decisions to rely on oneself or the decision aid we measured reliance when human and decision aid disagreed. Note that this could only be done in the 'human first' condition.

Perceived Reliability: After each experimental block participants estimated the reliability of both their own performance and that of the decision aid on a scale between 0 and 100% correct with steps of 10%. Relative trust in oneself is calculated by subtracting perceived reliability of the decision aid from perceived reliability of oneself. A positive value indicates that trust in oneself is higher than trust in the decision aid and a negative value that trust in oneself is lower.

Actual Reliability: Actual reliability of both the participant and the decision aid was measured during task execution. Reliability is defined as the percentage correct predictions (in rows 1, 2 and 3 in Figure 1 and rows 1 and 2 in Figure 2).

Understandability: Participants indicated on a Likert-scale from -3 to 3 (in steps of one) whether they thought the decision making process of themselves and that of the decision aid was understandable, where -3 indicated that it was completely not understandable and 3 meant that it was completely understandable. Relative understandability of oneself is calculated by subtracting understandability of the decision aid from understandability of oneself. A positive value indicates that understandability of oneself is higher than that of the decision aid and a negative value that understandability of oneself is lower.

Responsibility: Participants indicated on a Likert-scale from -3 to 3 (in steps of one) whether they felt responsible for the outcome of the task, where -3 meant they did not feel responsible at all and 3 meant they completely felt responsible.

Attribution of Unreliability: Participants indicated on a Likert-scale from -3 to 3 (in steps of one) whether unreliability in performance of oneself and decision aid is attributed to 'temporary factors', 'external factors' and 'uncontrollable factors', respectively (i.e., three times), where -3 meant that they thought that performance can absolutely not be attributed to those factors and 3 meant that they thought those factors absolutely play a role.

5 Results

5.1 Self Bias and Order of Advice (Hypothesis 1)

The percentage agreement with the decision aid in the 'decision aid first' condition ($M = 72.2\%$) did not differ from that in the 'human first' condition ($M = 72.5\%$), $t(155) = -0.19$, $p = .85$. In both conditions, on average, trust in the decision aid was higher than trust in oneself. But participants in the 'human first' condition perceived the decision aid to be 30% better than themselves compared to only 6% in the 'decision aid first' condition, $t(154) = 3.97, p < .01$. Whereas the perceived reliability of the decision aid's performance was only 7% higher in the 'decision aid first' condition ($M = 67.2\%$, $SD = 13.7$) than in the 'human first' condition ($M = 63.1\%$, $SD = 14.0$), $t(154) = 1.93, p = .055$. The perceived reliability of own performance was 30% lower in the 'human first' condition ($M = 48.5\%$, $SD = 17.4$) than in the 'decision aid first' condition ($M = 63.1\%$, $SD = 14.4$), $t(154) = -5.72$, $p < .01$. Differences in agreement with the decision aid between the 'human first' and 'decision aid first' conditions did not show up. This was probably because there was a significant difference between the conditions in relative trust. The lower percentage agreement that is expected in the 'human first' condition as a result of the self bias was not observed because participants in 'human first' condition on average thought their performance was 30% less than that of the decision aid.

Participants agree more with the decision aid when trust in oneself is lower than trust in the decision aid. This difference was significant in the 'human first' condition (left side of Figure 3), $t(76) = 2.15$, $p = .03$, but not in the decision aid first condition (right side of Figure 3), $t(76) = 0.97$, $p = .33$, probably because the difference in perceived reliability of own performance and that of the decision aid was less pronounced in the 'decision aid first' condition.

When trust in oneself is higher than trust in decision aid (right side of Figure 4), the percentage agreement with the decision aid in the 'human first' condition ($M = 68\%$) is lower than that in the 'decision aid first' condition ($M = 72\%$),

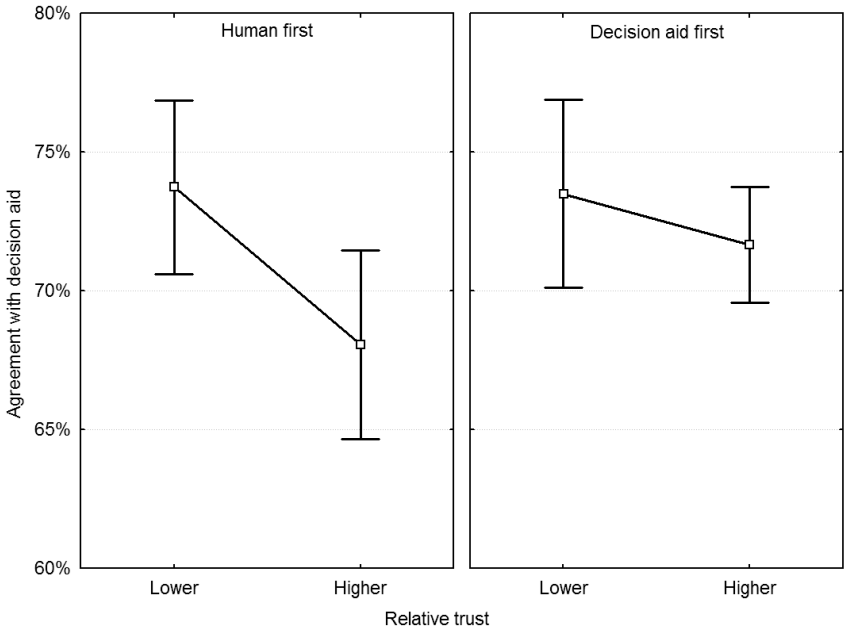


Figure 3. Agreement with the decision aid is higher when trust in oneself is lower than trust in the decision aid (lower relative trust), but only in the 'human first' condition.

$t(65) = 1.95$, $p = .056$. This indicates that a self bias is observed when participants first form their own opinion, but only when trust in oneself is higher than trust in the decision aid.

When trust in oneself is lower than trust in decision aid (left side of Figure 4), no difference in the percentage agreement between the 'human first' and 'decision aid first' condition is found, $t(87) = 0.1$, $p = .9$.

The results seem to be in agreement with Hypothesis 1: People rely more on themselves when they make their decision before receiving advice of a decision aid, but only when they trust themselves more than the decision aid. When trust in the decision aid exceeds trust in oneself no self bias effect is found.

5.2 Understandability of Underlying Reasoning (Hypotheses 2 and 3)

The data from the 'human first' condition was used to determine whether factors like relative understandability of underlying reasoning and feeling of responsibility in addition to relative trust in performance explain reliance on oneself or decision aid when in disagreement.

On average, participants estimated their own reliability to be 48.5% and that of the decision aid 63%. In other words, they thought the decision aid was 14.5% more reliable than themselves (and 30% more relatively speaking), $t(79) = -5.79$, $p < .01$. When the initial answer of the participant differed from the advice of the decision aid, participants relied for 52% on the decision aid and for 48% on themselves. This difference is not significant, $t(79) = -0.78$, $p = .44$. When participants would base their decisions to rely on oneself or decision aid on relative trust alone, one would expect them to at least rely 30% more often on the decision aid than on themselves. The results seem to be in agreement with Hypothesis 2: decision makers using decision aids rely less on the decision aid than would be expected based on relative trust alone. Participants did not rely more often on the decision aid when in

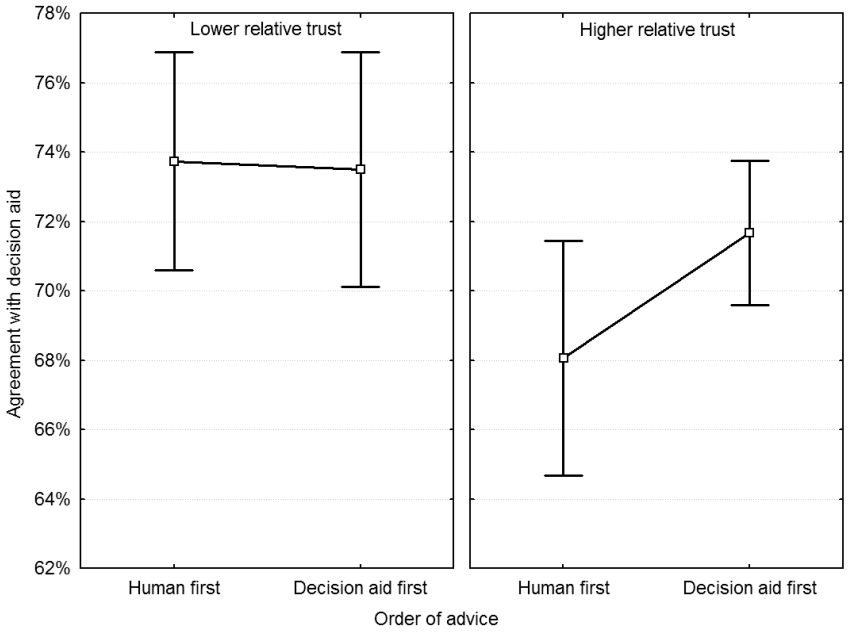


Figure 4. Agreement with the decision aid is higher in the 'decision aid first' condition, but only when trust in oneself is higher than trust in the decision aid (higher relative trust).

disagreement, although they perceived it to be 30% more reliable.

Also a regression analysis was performed in which reliance on oneself, when in disagreement with the advice of the decision aid, is regressed on relative trust, relative understandability of underlying processes and responsibility. The results indicate that relative trust has a unique contribution in predicting reliance ($\beta = .32$). The higher trust in oneself is relative to trust in the decision aid, the more participants also relied on their own decision rather than on the conflicting advice of the decision aid. These results are in agreement with the previously tested hypothesis that decision makers are less likely to accept conflicting advice when they perceive the advisor to be less reliable than themselves and vice versa.

As expected, on average, participants found their own decision making process to be understandable ($M = 0.64$, $SD = 1.4$), in contrast to that of the decision aid ($M = -0.93$, $SD = 1.32$), $t(79) = 7.07$, $p < .01$. Despite a correlation between relative trust and relative understandability ($r = .27$, $p < .05$), which is caused by a correlation between perceived reliability of oneself and understandability of oneself ($r = .37$, $p < .05$), results of the regression analysis indicate that relative understandability also contributes to predicting reliance on oneself ($\beta = .28$). The more understandable participants thought their own decision making process was compared to that of the decision aid, the more they relied on their own initial decision. These results are in agreement with our Hypothesis 3: decision makers rely less on conflicting advice when they perceive the advisor's reasoning to be cognitively less available and understandable than their own reasoning.

The results not only suggested that the more participants thought they understood how they formed their judgments, the more reliable they perceived themselves to be, but also that participants that were optimistic about their own performance, were also

optimistic about the performance of the decision aid ($r = .45$, $p < .05$).

5.3 Feeling of Responsibility (Hypothesis 4)

On average participants felt responsible for the accuracy of the final decision ($M = 1.25$, $SD = 1.11$). Differences in responsibility between individuals ranged between negative (-2) and absolutely positive (3). Results of the regression analysis indicate that responsibility also contributes to predicting reliance on oneself ($\beta = -.29$). The more responsible the participants felt for task outcome the more they relied on the conflicting advice of the decision aid rather than their own initial decision. These results are in agreement with Hypothesis 4: decision makers are more likely to accept (more reliable but) conflicting advice when they feel more responsible for the outcome of the decision.

Together relative trust, relative understandability and responsibility explain 38% of the variance in reliance. Based on the magnitudes of the beta-coefficients, the squared partial and semi-partial correlations (see Table I), the relative contribution of these factors seems to differ little.

5.4 Accuracy of Perceived Reliability (Hypothesis 5)

Perceived Reliability of Oneself: Results show that some participants underestimated their reliability, while others overestimated their reliability (Figure 5), but averaged over two blocks perceived reliability of own performance was 4% lower than it actually was, $t(79) = -2.53$, $p < .01$.

For the first block participants underestimated their performance with 5%, $t(39) = -2.16$, $p < .05$. But underestimation was not statistically significant in the second block, $t(39) = -1.46$, $p > .05$ (Figure 6). Correlations between perceived reliability and actual reliability of own performance increased from $r = .42$, $p < .05$ in the first to $r = .51$, $p < .05$ in the second block. We also found that estimations of own reliability improve after time and that underestimation seems to disappear after time.

Perceived Reliability of the Decision Aid: Most participants underestimated the reliability of the decision aid, but both pessimists that over-weighed errors as well as optimists that under-weighed errors were found (Figure 7). Averaged over two blocks the perceived reliability of the decision aid was 7% lower than it actually was, $t(79) = -4.41$, $p < .01$.

For the first block participants underestimated the performance of the decision aid for 7%, $t(39) = -2.47$, $p < .05$ and for 8% in the second block, $t(39) = -3.79$, $p < .01$ (Figure 6).

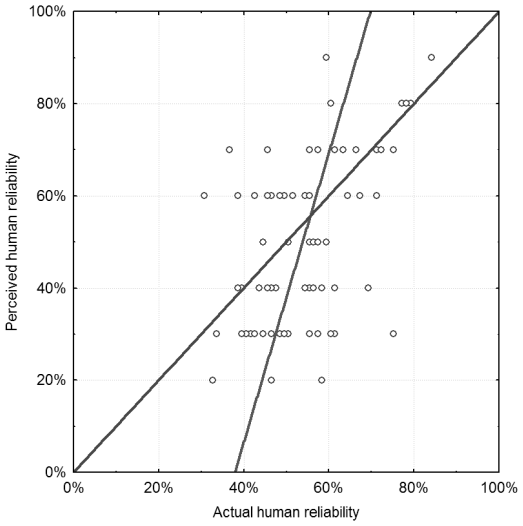


Figure 5. Calibration human reliability.

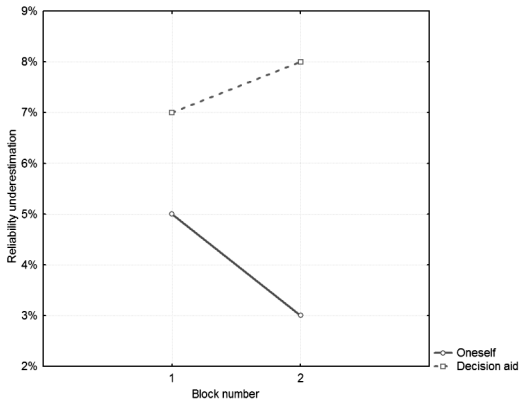


Figure 6. Effects of learning on estimation of reliability of oneself and decision aid.

Table I
REGRESSING RELIANCE ON RELATIVE TRUST, RELATIVE UNDERSTANDABILITY AND RESPONSIBILITY.

	Regression coefficients		Squared partial correlations	Squared semi-partial correlations
	Beta	Std error		
Relative Trust	.32*	.098	.12	.09
Relative Understandability	.28*	.097	.10	.07
Responsibility	−.29*	.094	.11	.08

* $p < .01$

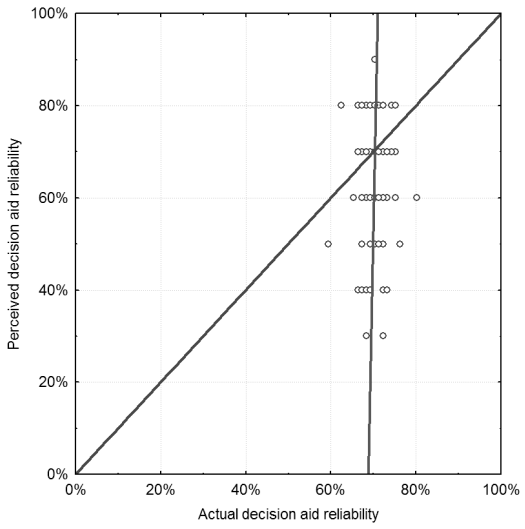


Figure 7. Calibration decision aid reliability.

In the first block the standard deviation of perceived reliability was slightly higher ($SD = 14.30$) than in the second block ($SD = 13.53$).

These results are in agreement with Hypothesis 5: On average reliability of own performance and that of the decision aid is underestimated when people are provided with feedback about performance.

5.5 Attribution Bias (Hypotheses 6 and 7)

The above results are also in agreement with Hypothesis 6: underestimation of the reliability of the decision aid is expected to be more prevalent and more persistent than underestimation of reliability of oneself. In sum, we found underestimation for both oneself and decision aid, but underestimation was higher for the decision aid. Also underestimation of own reliability decreased after practice; that of the decision aid did not.

On average, unreliability of the decision aid is less attributed to temporary factors ($M = 0.05$) than own unreliability ($M = 0.41$), $t(79) = 2.02$, $p < .05$. Unreliability of the decision aid is also less attributed to uncontrollable factors ($M = -0.85$), than own unreliability ($M = -0.26$), $t(79) = 2.92$, $p < .05$. However, both own unreliability ($M = -0.79$) and that of the decision aid ($M = -1.09$) was not attributed to external factors, no difference between self and decision aid was found, $t(77) = 1.66$, $p > .05$. The above results are partly in agreement with Hypothesis 7: unreliability of the decision aid is less attributed to temporary and uncontrollable causes, but like own unreliability is not less attributed to external causes.

6 CONCLUSION

A self bias is expected when the effort to rely on one's own judgment is perceived to be lower than to change one's mind and to accept the conflicting advice of a decision aid. This tendency is expected when one's own judgment is cognitively more 'available', for instance because it is formed before rather than after receiving advice. Self bias can only occur when given advice conflicts with one's initially held beliefs. The results have shown that the self bias can be observed and that people disagree more with a decision aid when they express their decision before rather than after receiving advice. The results also show that this is only the case when decision makers trust themselves more than the decision aid (Hypothesis 1). No self bias was found when trust in the decision aid exceeded trust in oneself. We therefore argue that in existing

frameworks of automation use, the notion of automation bias needs to be complemented with that of the self bias. Whether self biases lead to desirable outcomes or not, depends on whether perceptions of reliability of one's own performance and that of the decision aid are appropriate. When people wrongly think they perform better than the decision aid, self reliance can result in undesirable outcomes.

There is reason to believe that decision makers do not sufficiently rely on advice of decision aids. Our results show that decision makers rely less on the decision aid than would be expected based on relative trust in performance reliability alone. Participants did not rely more often on the decision aid when in disagreement, although they perceived it to be 30% more reliable (Hypothesis 2). Our results suggest that decision makers rely less on conflicting advice because they perceive the advisor's reasoning to be cognitively less available and understandable than their own reasoning (Hypothesis 3). Together with relative trust, relative understandability and responsibility explain 38% of the variance in reliance.

People who felt more responsible for the task outcome relied more on conflicting advice than people who feel less responsible (Hypothesis 4). It seems that when people feel more responsible that they are more willing to invest the mental effort that is required to let go their initial decision and accept conflicting advice of the decision aid.

Perceived reliability of both oneself and decision aid is underestimated when feedback about performance is provided (Hypothesis 5) and it seems that negative experiences have a greater influence than do positive experiences. Since relative trust is based on the difference between perceived reliability of oneself and decision aid, as long as the degree of underestimation does not differ between oneself and decision aid, no problems with the decision to rely on advice is expected. However, when the magnitude or direction of underestimation differs, inappropriate reliance decisions may be the result. Our results suggest however that the underestimation of the reliability of the decision aid is more and more persistent (Hypothesis 6).

It seems users are less forgiving and less optimistic about the performance of the decision aid, even though on the group level it performs 30% more reliable, probably because errors are less attributed to temporary and uncontrollable causes (Hypothesis 7).

A note on the scalability of this research. The reason for using a pattern learning task in this study is that it can be controlled very well and hypotheses can be tested quite precisely. More realistic settings in which the results of this study are expected to scale to are for example all tasks that incorporate decision making based on advice from different agents (man or machine). The reliance decisions studied in this paper can be seen as largely independent of the task at hand and therefore the drawn conclusions are expected to scale to these more realistic tasks and more ecologically relevant.

Finally some decision aid design implications of the present research. Appropriate reliance on decision aids is not guaranteed when only focusing on optimizing the reliability of decision aids. There are several things one could do in the design phase of a decision aid. First of all, give people feedback about their own individual performance, that of the decision aid and team performance, but correct for the bias that negative information is given more weight. This feedback can improve the calibration of trust in oneself and decision aid and therefore stimulate appropriate reliance. Secondly, by providing advice after, rather than before, more knowledge is brought to the task. Such a design is not focused on reducing workload by automation, but focused on human-machine collaboration with the goal of increasing accuracy and resilience. Receiving advice afterward may also increase confidence of the decision maker when both human and system agree or make people think twice when both disagree. The designer should aim at reducing the effort to rely on oneself and decision aid to make human-computer collaboration more flexible. One should make the reasoning of the decision aid available and understandable in the human-computer interface. Also, make people feel accountable for the outcomes of the human-computer team. Hold people responsible for quality of outcome of the human-computer team. Finally, one should control for the attribution of errors. For instance by making sources of error transparent or by making operators aware of their biases in attribution. The idea is that providing information regarding why the automation might be mistaken increases trust (Dzindolet et al., 2003).

ACKNOWDLEGMENTS

This research was partly funded by the Royal Netherlands Navy under programme numbers V206 and V929. The authors are grateful to Jan Maarten Schraagen, Jasper Lindenberg and Tibor Bosse for their comments and suggestions.

REFERENCES

Alba, J. W. and Hutchinson, J. W. (2000). Knowledge calibration: What consumers know and what they think they know. *Journal of Consumer Research*, 27:123–156.

Bonaccio, S. and Dalal, R. S. (2006). Advice taking and advice giving in decision making: an integrative review of the literature. Working paper.

Bowers, C. A., Oser, R. L., Salas, E., and Cannon-Bowers, J. A. (1996). Team performance in automated systems. In Parasuraman, R. and Mouloua, M., editors, *Automation and human performance: Theory and applications*, pages 243–263. Lawrence Erlbaum Associates, Inc., Mahwah, NJ.

Brenner, L. A., Koehler, D. J., Liberman, V., and Tversky, A. (1996). Overconfidence in probability and frequency judgments: A critical examination. *Organizational Behavior and Human Decision Processes*, 65:212–219.

Dzindolet, M. T., Beck, H. P., Pierce, L. G., and Dawe, L. A. (2001). A framework of automation use. Technical Report ARL-TR-2412, Army Research Laboratory, Aberdeen Proving Ground, MD.

Dzindolet, M. T., Peterson, S. A., Pomransky, R. A., Pierce, L. G., and Beck, H. P. (2003). The role of trust in automation reliance. *International Journal of Human Computer Studies*, 58(6):697–718.

Dzindolet, M. T., Pierce, L. G., Beck, H. P., and Dawe, L. A. (2002). The perceived utility of human and automated aids in a visual detection task. *Human Factors*, 44:79–94.

Gao, J. and Lee, J. D. (2006). Extending decision field theory to model operator's reliance on automation in supervisory control situations. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 36(5):943–959.

Gilbert, D. T. and Malone, P. S. (1995). The correspondence bias. *Psychological Bulletin*, 117:21–38.

Guerlain, S., Smith, P. J., Obradovich, J. H., Rudmann, S., Strohm, P., Smith, J. W., Svirbely, J., and Sachs, L. (1999). Interactive critiquing as a form of decision support: An empirical evaluation. *Human Factors*, 41(1):72–89.

Lee, J. D. and See, K. A. (2004). Trust in automation: Designing for appropriate reliance. *Human Factors*, 46(1):50–80.

Madhavan, P. and Wiegmann (2007). Similarities and differences between human-human and human-automation trust: An integrative review. *Theoretical Issues in Ergonomics Science*, 8(4).

May, R. S. (1987). *Calibration of subjective probabilities: A cognitive analysis of inference processes in overconfidence*. Peter Lang, Frankfurt. in German.

May, R. S. (1988). Overconfidence in overconfidence. In Chaikan, A., Kindler, J., and Kiss, I., editors, *Proceedings of the 4th FUR Conference*, Dordrecht. Kluwer.

Merritt, S. M. and Ilgen, D. R. (2008). Not all trust is created equal: Dispositional and history-based trust in human-automation interactions. *Human Factors*, 50:194–210.

Miller, C. A. (2002). Definitions and dimensions of etiquette. Technical Report FS-02-02, American Association for Artificial Intelligence, Menlo Park, CA.

Mosier, K. L. and Skitka, L. J. (1996). Human decision makers and automated decision aids: Made for each other? In Parasuraman, R. and Mouloua, M., editors, *Automation and human performance: Theory and applications*, pages 201–220. Lawrence Erlbaum Associates, Inc., Mahwah, NJ.

Parasuraman, R. and Riley, V. A. (1997). Humans and automation: Use, misuse, disuse, abuse. *Human Factors*, 39:230–253.

Silverman, B. (1992). Survey of expert critiquing systems: Practical and theoretical frontiers. *CACM*, 35(4):106–127.

Skitka, L. J., Mosier, K. L., and Burdick, M. (1999). Does automation bias decision-making? *International Journal of Human-Computer Studies*, 51(5):991–1006.

Sniezek, J. A. and Buckley, T. (1995). Cueing and cognitive conflict in judge-advisor decision making. *Organizational Behavior and Human Decision Processes*, 62:159–174.

Thomas, L. C. and Rantanen, E. M. (2006). Human factor issues in implementation of advanced aviation technologies: A case of false alerts and cockpit displays of traffic information. *Theoretical Issues in Ergonomics Science*, 7:501–523.

Tversky, A. and Kahneman, D. (1973). Availability: A heuristic for judging frequency and probability. *Cognitive Psychology*, 5:207–232.

Tversky, A. and Koehler, D. J. (1994). Support theory: A nonextensional representation of subjective probability. *Psychological Review*, 101:547–567.

van Dongen, K. and van Maanen, P.-P. (2006). Under-reliance on the decision aid: A difference in calibration and attribution between self and aid. In *Proceedings of the Human Factors and Ergonomics Society's 50th Annual Meeting*.

Weiner, B. (1986). *An attributional theory of motivation and emotion*. Springer-Verlag, New York.

Wickens, C. D. and Dixon, S. R. (2007). The benefits of imperfect diagnostic automation: A synthesis of the literature. *Theoretical Issues in Ergonomics Science*, 8(3):201–212.

Wiegmann, D. A., Rich, A., and Zhang, H. (2001). Automated diagnostic aids: the effects of aid reliability on user's trust and reliance. *Theoretical Issues in Ergonomics Science*, 2:352–367.

Yaniv, I. and Kleinberger, E. (2000). Advice taking in decision making: Egocentric discounting and reputation formation. *Organizational Behavior and Human Decision Processes*, 83:260–281.

Chapter 6

Aiding Human Reliance Decision Making Using Computational Models of Trust

This chapter appeared as (van Maanen et al., 2007a).

Aiding Human Reliance Decision Making Using Computational Models of Trust

Peter-Paul van Maanen[*†], Tomas Klos[‡] and Kees van Dongen[*]

[*] Department Human in Command, TNO Human Factors
P.O. Box 23, 3769 ZG Soesterberg, The Netherlands
Email: {peter-paul.vanmaanen, kees.vandongen}@tno.nl
[†] Department of Artificial Intelligence, Vrije Universiteit Amsterdam
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
[‡] Dutch National Research Institute for Mathematics and Computer Science (CWI)
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
Email: tomas.klos@cwi.nl

***Abstract*—This paper involves a human-agent system in which there is an operator charged with a pattern recognition task, using an automated decision aid. The objective is to make this human-agent system operate as effectively as possible. Effectiveness is gained by an increase of appropriate reliance on the operator and the aid. We studied whether it is possible to contribute to this objective by, apart from the operator, letting the aid as well calibrate trust in order to make reliance decisions. In addition, the aid's calibration of trust in reliance decision making capabilities of both the operator and itself is also expected to contribute, through reliance decision making on a meta-level, which we call meta-reliance decision making. In this paper we present a formalization of these two approaches: a reliance (RDMM) and meta-reliance decision making model (MetaRDMM), respectively. A combination of laboratory and simulation experiments shows significant improvements compared to reliance decision making solely done by operators.**

1 Introduction

Human-aid cooperation in complex domains, such as aviation, nuclear power, or health care, is becoming increasingly common. The idea of this is that the performance of humans in closer cooperation with decision aids (agents), and vice versa, perform better than humans or decision aids working separately, without taking the other into account. Although this performance benefit is often observed in human-aid teams, cooperation effectiveness of the decision aid is not always fully realized.

In recent work (van Dongen and van Maanen, 2006; van Maanen and van Dongen, 2005) a human-aid team was studied where a human operator, charged with a pattern recognition task, was supported by an automated decision aid. The objective of the task was to make this human-aid team operate as effectively as possible. It turned out that in many occasions the operator made wrong reliance decisions and therefore effectiveness decreased.

Ideally humans rely on their own decisions when these are best and rely on the decision aid's when those are best. But operators cannot be expected to base their reliance decisions on comparisons of true reliabilities of themselves and those of the decision aids. Rather, perceived reliabilities are used which, unfortunately, are usually imperfectly calibrated to true reliabilities, even after practice (van Dongen and van Maanen, 2006). It is often found that humans rely either too much or too little on decision aids or themselves (Parasuraman and Riley, 1997; Skitka et al., 1999; Dzindolet et al., 1999; van Dongen and van Maanen, 2006).

People use relative trust to decide whom to rely on (Moray et al., 2000). Trust is defined as the attitude that an agent will help achieve an individual's goals in a situation characterized by uncertainty and vulnerability (Lee and See, 2004). Trust can refer to the advice of another agent or to one's own judgment. Trust, like the perceptions of reliability on which it is based, is a covert or cognitive state (Falcone and Castelfranchi, 2001).

Perceptions of reliability may be prone to systematic and random error. One such error is over-trust: humans may overestimate their own performance or that of the aid. Humans are for instance known to overestimate the number of tasks they can complete in a given period of time (Buehler et al., 1994). Another error is under-trust: humans may underestimate their own performance or that of the aid. Concerning the perception of the aid's performance, in (Wiegmann et al., 2001; van Dongen and van Maanen, 2006) it was for instance found that the reliability of decision aids is often underestimated. When the direction or magnitude of such errors differ between self and aid, this could lead to inappropriate reliance decisions: Under-reliance or over-reliance may be the result.

Because the aid is unaffected by cognitive biases, like humans are, the first question raised in this paper is whether it is possible to let the aid make more accurate trust assessments, and therefore reliance decisions, than the operator. In that case, reliance decision making done by the aid is expected to lead to a decrease of over- and under-reliance.

Nonetheless, the transparent character of the operator's own motivation for his performance may result in a substantial amount of occasions in which humans make better reliance decisions than aids. In these cases, the suggested reliance decision making completely done by the aid does not result in an optimal performance. The second question therefore raised is whether it is possible to let the aid make even more accurate reliance decisions when based on a prediction if such situations are at hand and then the decision is made to rely on the operator if that is more appropriate. This type of decision making is on a meta-level and therefore is called *meta*reliance decision making. It is expected to result in a further decrease of over- and under-reliance.

This paper is composed of several sections addressing the above two questions. First, in Section 2 it is shown how an extension of the task environment from (van Dongen and van Maanen, 2006) is used as a base for studying the effectiveness of aiding human reliance decision making. Decision aid design and the formalization of the reliance decision making models used by the aid, i.e., a *reliance (RDMM)* and *metareliance decision making model (MetaRDMM)*, respectively, are presented in Section 3. Section 4 describes the method of the experiment and simulation done. The results in terms of model performance by comparison with operator performance from (van Dongen and van Maanen, 2006) are presented in Section 5. Section 6 ends this paper with some conclusions and suggestions for further research.

2 Task Environment

For the experiment described in (van Dongen and van Maanen, 2006) participants read a story about a software company interested in evaluating the performance of their adaptive software before applying it to more complex tasks on naval ships. Participants were asked to perform a pattern recognition task with advice of a decision aid and were instructed to maximize the number of correct answers by either relying on their own or the decision aid's predictions.

The interface of the task contained 4 rows. Each row consisted of a progress bar, buttons numbered 1, 2, and 3, and a phase description. In Phase 1 the operator had to predict which button to push, based on what they thought the pattern was. In Phase 2 the aid had to do the same. In Phase 3 the operator again had to decide which button to push, this time also taking the prediction of the aid into account, which required the operator also to make a reliance decision. In the final phase feedback was given on what button was correct. Each experiment contained 101 trials, each consisting of these four phases.

In order to support the operator in making reliance decisions the above rows were extended to a total of six (see example interaction in Figure 1), which means two phases were added: In Phase 4 the aid had to make a reliance decision similar as the operator's in Phase 3. In Phase 5 the aid had to decide when to follow the reliance decision of the operator and when its own. These kind of decisions are called *meta*reliance decisions. After this, Phase 6 was the feedback phase.

In Figure 1 the following scenario is shown: the operator predicts number 3 (Phase 1), the aid number 2 (Phase 2), then the operator wants to rely on himself (Phase 3), the aid also relies on itself (Phase 4), then the aid decides to *meta*rely on itself again (Phase 5), and finally the feedback shows

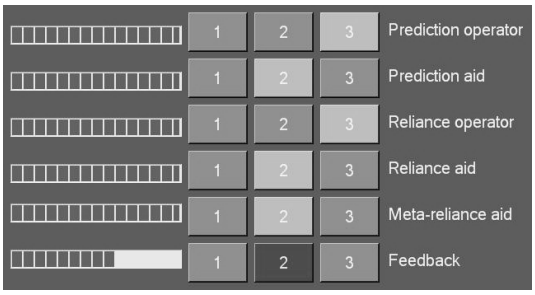


Figure 1. Example operator-aid interaction. The rows represent different phases.

this was the appropriate decision (Phase 6). Both interpret the outcome and go on to the next trial.

Note that no other support than mentioned above is given to the operator (e.g., no correct answer history is kept for the operator and the operator was not allowed to write things down). Feedback is based on a predefined but then partly randomized sequence of the numbers 1, 2, and 3. The predictions of the aid were also predefined. Each participant got a comparable but different sequence. See Section 4 for more details.

3 Decision Aid Design

In this section the design of the aid is described in terms of how the aid's decisions in Phases 4 (RDMM) and 5 (MetaRDMM) are made (see end of Section 4 for details on Phase 2), building on (Klos and La Poutré, 2006; van Maanen and van Dongen, 2005). In these phases the aid estimates and compares the task-related (prediction) and reliance decision making capabilities, respectively, of the operator and itself. The idea is to let the aid estimate its trust in the operator's and its own prediction and reliance decision making capabilities each time that feedback is given in Phase 6. As a model, we use a Beta probability density function (pdf) over the different values that the agents' (operator's or aid's) respective capabilities can have. Upon receiving the feedback in Phase 6, the aid uses Bayes' rule to update its estimations (for a generalization to the Dirichlet distribution see Krukow, 2006) (Gelman et al., 2004; Jøsang and Ismail, 2002; Klos and La Poutré, 2006).

From the perspective of the aid, each agent's behavior can be seen as a sequence of Bernouilli trials, governed by a bias or probability of the outcome 'success' in each trial, called θ_a^x for $x \in \{\texttt{prediction}, \texttt{reliance}\}$ and $a \in \{\texttt{operator}, \texttt{aid}\}$. It is this probability that the aid needs to estimate for the two possible values of both x and a. For $x =$ prediction, this yields two values for RDMM, and for $x =$ reliance, it yields two values for MetaRDMM. In the remainder of this section we drop the sub- and superscripts a and x.

The probability of n successes in N Bernouilli trials ($0 \leq n \leq N$) is given by the Binomial probability mass function

$$p(n|\theta) = \binom{N}{n}\theta^n(1-\theta)^{N-n}. \tag{1}$$

This also gives the Binomial likelihood of θ, when interpreted as a function of the second argument θ with n held fixed. This likelihood may be used to update the posterior probability $p(\theta|n)$, using Bayes' rule:

$$p(\theta|n) = \frac{p(n|\theta)p(\theta)}{p(n)}. \tag{2}$$

The Beta pdf is a conjugate prior for the Binomial likelihood, which means that if it is used as the prior $p(\theta)$ in Eq. 2, the posterior $p(\theta|n)$ is again a Beta pdf. The Beta pdf is the following:

$$\mathrm{Beta}(\theta|r,s) = \frac{1}{\beta(r,s)}\theta^{r-1}(1-\theta)^{s-1}, \tag{3}$$

for $0 \leq \theta \leq 1$ and $s, r > 0$, where $\beta(r,s)$ is the beta function, and s and r are the number of successes and failures, respectively. [1] The expected value of the Beta distribution is $E(\theta) = \frac{r}{r+s}$.

As explained above, the posterior distribution is still a Beta distribution (disregarding the normalization factor in the denominator of Bayes' rule, since it does not depend on θ):

$$\overbrace{p(\theta|n,N,r,s)}^{\text{posterior}} \propto \overbrace{\left[\theta^n(1-\theta)^{N-n}\right]}^{\text{likelihood (see Eq. 1)}} \overbrace{\left[\theta^{r-1}(1-\theta)^{s-1}\right]}^{\text{prior (see Eq. 3)}}$$
$$\propto \theta^{n+r-1}(1-\theta)^{N-n+s-1},$$

[1] The beta function is

$$\beta(r,s) = \frac{\Gamma(r)\Gamma(s)}{\Gamma(r+s)},$$

where $\Gamma(x) = (x-1)!$ is the Gamma function, where x is a positive integer.

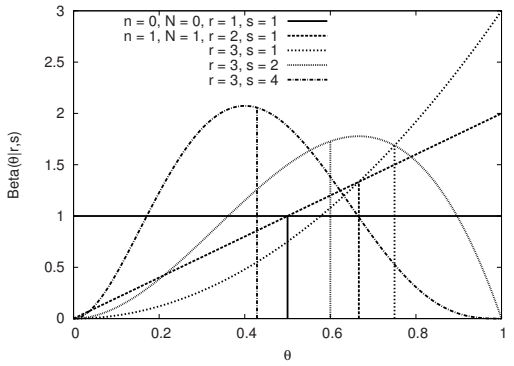


Figure 2. The Beta pdf of θ for different values of r and s.

with expected value $E(\theta) = \frac{r+n}{r+s+N}$. In effect, one simply adds the new counts of successes (n) and failures ($N-n$) to the old values of the parameters of the Beta distribution r and s, respectively, and obtains a new distribution with parameters r' and s'.

In the context of trust models, an agent i's trust $\tau_i(j)$ in another agent j's capabilities or intentions, is usually calculated as the expected value of the beta function $\beta(u+1, v+1)$, where u and v are the current counts of positive and negative experiences i has had with j. In the absence of such experiences, the values $r = s = 1$ are typically used for binary outcomes, yielding a uniform prior, and an expected value of 0.5 for the value estimated to govern j's behavior. Updating this uniform prior with positive an negative evidence u and v, respectively, yields $E(\theta) = \frac{u+1}{u+v+2}$. Figure 2 gives the shape of the Beta probability density function of θ given different amounts of evidence, where the expected values of θ are indicated by vertical lines. Because we have 3 possible outcomes in each phase, we initialize the prior as $p = \frac{1}{3}$, by setting $r = 1$ and $s = 2$. Furthermore, we discount old evidence (Jøsang and Ismail, 2002), by using a discount factor $0 \leq \lambda \leq 1$, with which old evidence is multiplied before new evidence is added.

For each trial, when the two agents' predictions or reliances differ, (Meta)RDMM selects the prediction (in Phase 4) or reliance (in Phase 5) made by the most highly trusted agent, using four updated trust values $\tau^x_{\texttt{aid}}(a)$ of the aid. In Figure 3 the aid's trust dynamics for an arbitrary operator are shown. For trial 14, for instance, the phase outcomes are similar as in the scenario shown in Figure 1: For this trial, the operator predicted 3, the aid 2, the operator relied on himself, and the correct number was 2 (Phases 1–3, 6). Because the operator prediction trust is lower than the aid prediction trust ($\tau^{\texttt{prediction}}_{\texttt{aid}}(\texttt{operator}) < \tau^{\texttt{prediction}}_{\texttt{aid}}(\texttt{aid})$), the aid relied on itself (Phase 4), and because the operator reliance trust was lower than the aid reliance trust ($\tau^{\texttt{reliance}}_{\texttt{aid}}(\texttt{operator}) < \tau^{\texttt{reliance}}_{\texttt{aid}}(\texttt{aid})$), the aid also *meta*relied on itself (Phase 5).

4 Method

4.1 Participants

The experimental data related to the input of the operator (Phases 1 and 3) were taken from the experiments described in (van Dongen and van Maanen, 2006). Forty three Dutch university students (16 female, 18–38 yrs, $M = 23$ yrs) participated in the experiment. Participants were paid € 35 for their participation. To control for learning effects, each participant performed the same experiment twice. This means there were a total of 86 experiments, each containing 101 trials. The decision aid was *simulated* offline to be aiding these participants as described in Section 2.

4.2 Design

Performances for three phases were calculated: the operator's reliance phase (OperatorRDM), the aid's reliance phase (RDMM), and the aid's meta-reliance phase (MetaRDMM), i.e., Phases 3–5. Only those trials were interesting in which either the operator or (exclusive or) the aid made a correct prediction or reliance decision. This is due to the fact that, in the case of prediction and reliance consensus and in the situation where neither operator nor aid is correct in their prediction or reliance, comparison of aid and operator performance is uninformative.[2] The independent variables for each performance measure were operator and aid prediction accuracy (for Phases 1 and 2), which are described below in more detail.

Operator prediction accuracy was manipulated by varying the difficulty of predicting a predefined

[2]Although it appears that in the experiments 0.64% of the trials participants decided not to, or were too late to, rely on prediction consensus, it had no significant influence on the present results.

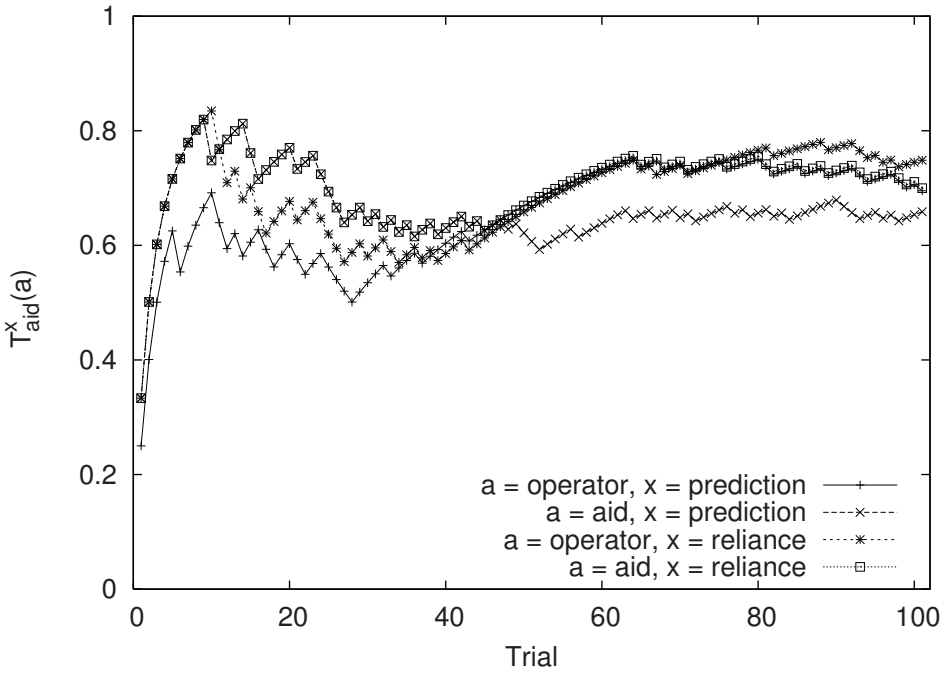


Figure 3. Example of different trust dynamics for an arbitrary operator.

sequence of the numbers 1, 2, and 3. The order of the predefined sequence determined the order of the given feedback in Phase 6, which was the only source for the participants to learn the sequence. The predefined sequence was a repeated, but randomized, pattern of length 5. Note here that participants did not know they were subject to identification of a (randomized) repeated sequence. Due to the fact that humans tend to see patterns in noise and because of a convincing story told in the beginning of the experiment, they rather thought it was a sequence dependent on certain complex patterns still to be discovered by them. This has also been confirmed by a post-experimental questionnaire.

Aid prediction accuracy manipulation was based on randomization of the above mentioned randomized predefined sequence. The accuracy of the aid was set on average at 70% ($SD = 3\%$), which is similar to the expected operator prediction accuracy. This was done to make reliance decision making nontrivial for the operator.

5 Results

Based on the experiments it is found that on average, for each participant, in 47.64% ($SD = 6.23\%$) of all trials either the operator ($M = 34.19\%$, $SD = 12.44\%$) or the aid ($M = 65.81\%$, $SD = 12.44\%$) predicted correctly. These last two averages differ substantially from 0 ($N = 48$, $p < .001$), which suggests that optimal performance is not reached simply by relying only on the aid or operator. For the mentioned trials, the performances (percentages

correct) of OperatorRDM ($M = 58.65\%$, $SD = 9.79\%$) and RDMM ($M = 66.38\%$, $SD = 10.43\%$) are shown in Figure 4 (empty bars). The RDMM results show a significant improvement compared to OperatorRDM ($t = 4.98$, $p = 0.00$).

On average, for each participant, in 22.04% ($SD = 9.88\%$) of all trials either the operator ($M = 40.85\%$, $SD = 18.28\%$) or the aid ($M = 59.15\%$, $SD = 18.28\%$) relied correctly. These last two averages differ substantially from 0 ($N = 22$, $p < .001$), which suggests that reliance decision making completely done by the aid does not result in an optimal performance. For the mentioned trials, the performances of OperatorRDM, RDMM, and MetaRDMM ($M = 59.80\%$, $SD = 16.81\%$) are shown in Figure 4 (pattern bars). The MetaRDMM results show a significant improvement compared to OperatorRDM ($t = 7.03$, $p < .001$) and an insignificant improvement compared to RDMM ($t = 0.24$, $p = .81$). There is no significant difference between the two experiments per participant. Hence, there are no measurable learning effects.

6 Conclusion

The general goal of this work is to develop concepts that improve performance of human-aid teams. Improvement is reached by aiding human reliance decision making through the usage of computational models of trust. Our results showed significant results in which decision models RDMM and MetaRDMM outperform human reliance decision making capabilities. The participants may have performed worse than (Meta)RDMM because of limited attentional and memory resources and biases in weighing successes and failures of both themselves and the aid.

As was expected, the results still show a substantial amount of occurrences in which humans make better reliance decisions than aids. MetaRDMM tries to take advantage of this. Although our results show that MetaRDMM also outperforms human participants, a significant improvement compared to RDMM was not found. The first research question raised in this paper can thus be answered with yes, but the answer for the second remains a challenge for further research. Results may differ if the experiment is redone using the extended task described in this paper. One of the positive effects MetaRDMM might imply is a lower human performance degradation, and thus a stronger advantage to RDMM.

It is expected that in real world settings both human reliance decision making and the opportunities for support will be different. Humans, for instance, use additional cues for calibrating trust. Also feedback is often not immediately available, is not always accurate, or complete. The application of the presented concepts and models in real world settings must therefore also be subject to further exploration.

Acknowledgments

This research is partly funded by the Royal Netherlands Navy under progr. nr. V206 and by the Dutch government (SENTER) under proj. nr. TSIT2021.

References

Buehler, R., Griffin, D., and Ross, M. (1994). Exploring the "planning falacy": Why people underestimate their task completion times. *Journal of Personality and Social Psychology*, 67:366–381.

Dzindolet, M. T., Pierce, L. G., Beck, H. P., and Dawe, L. A. (1999). Misuse and disuse of automated aids. In *Proceedings of the Human Factors Society 43rd Annual Meeting*, pages 339–343, Santa Monica, CA.

Falcone, R. and Castelfranchi, C. (2001). Social trust: a cognitive approach. *Trust and deception in virtual societies*, pages 55–90.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004). *Bayesian Data Analysis*. Chapman & Hall/CRC, second edition.

Jøsang, A. and Ismail, R. (2002). The Beta reputation system. In *Proc. 15^{th} Bled Electronic Commerce Conference*, Bled, Slovenia.

Klos, T. B. and La Poutré, H. (2006). A versatile approach to combining trust values for making binarydecisions. In *Trust Management*, volume 3986 of *Lecture Notes in Computer Science*, pages 206–220. Springer.

Krukow, K. (2006). *Towards a Theory of Trust for the Global Ubiquitous Computer*. PhD thesis, University of Aarhus.

Lee, J. D. and See, K. A. (2004). Trust in automation: Designing for appropriate reliance. *Human Factors*, 46(1):50–80.

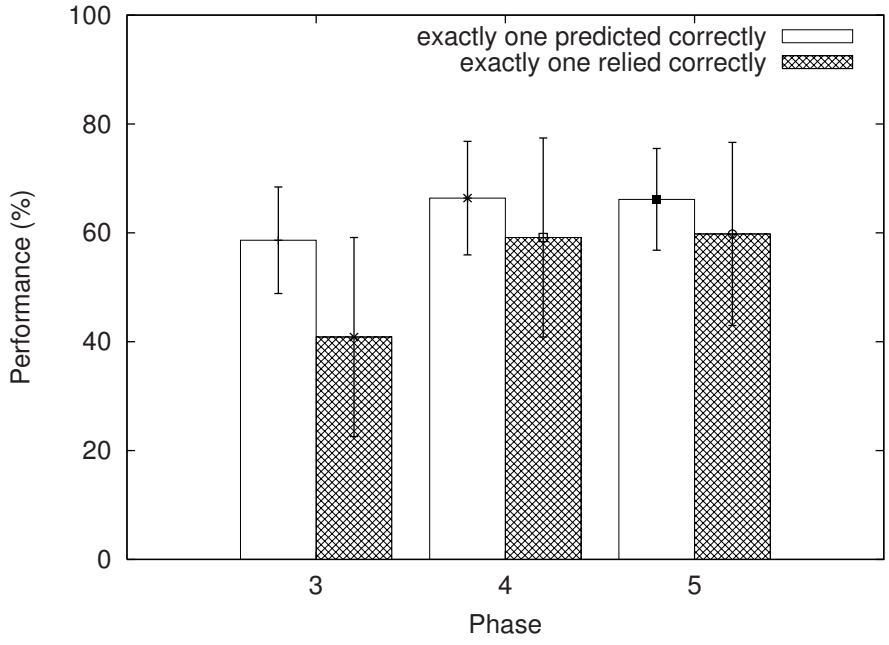


Figure 4. OperatorRDM (Phase 3), RDMM (Phase 4), and MetaRDMM (Phase 5) performances.

Moray, N., Inagaki, T., and Itoh, M. (2000). Adaptive automation, trust, and self-confidence in fault management of time-critical tasks. *Journal of Experimental Psychology: Applied*, 6(1):44–58.

Parasuraman, R. and Riley, V. A. (1997). Humans and automation: Use, misuse, disuse, abuse. *Human Factors*, 39:230–253.

Skitka, L. J., Mosier, K. L., and Burdick, M. (1999). Does automation bias decision-making? *International Journal of Human-Computer Studies*, 51(5):991–1006.

van Dongen, K. and van Maanen, P.-P. (2006). Under-reliance on the decision aid: A difference in calibration and attribution between self and aid. In *Proceedings of the Human Factors and Ergonomics Society's 50th Annual Meeting*.

van Maanen, P.-P. and van Dongen, K. (2005). Towards task allocation decision support by means of cognitive modeling of trust. In Castelfranchi, C., Barber, S., Sabater, J., and Singh, M., editors, *Proceedings of the Eighth International Workshop on Trust in Agent Societies (Trust 2005)*, pages 168–77.

Wiegmann, D. A., Rich, A., and Zhang, H. (2001). Automated diagnostic aids: the effects of aid reliability on user's trust and reliance. *Theoretical Issues in Ergonomics Science*, 2:352–367.

Chapter 7

Validation and Verification of Agent Models for Trust: Independent Compared to Relative Trust

Validation and Verification of Agent Models for Trust: Independent Compared to Relative Trust

Mark Hoogendoorn*, Syed Waqar Jaffry* and Peter-Paul van Maanen*†

* Department of Artificial Intelligence, Vrije Universiteit Amsterdam
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
Email: {mhoogen, swjaffry}@cs.vu.nl
† Department of Cognitive Systems Engineering, TNO Human Factors
P.O. Box 23, 3769 ZG Soesterberg, The Netherlands
Email: peter-paul.vanmaanen@tno.nl

***Abstract*—In this paper, the results of a validation experiment for two existing computational trust models describing human trust are reported. One model uses experiences of performance in order to estimate the trust in different trustees. The second model carries the notion of relative trust. The idea of relative trust is that trust in a certain trustee not solely depends on the experiences with that trustee, but also on trustees that are considered competitors of that trustee. In order to validate the models, parameter adaptation has been used to tailor the models towards human behavior. A comparison between the two models has also been made to see whether the notion of relative trust describes human trust behavior in a more accurate way. The results show that taking trust relativity into account indeed leads to a higher accuracy of the trust model. Finally, a number of assumptions underlying the two models are verified using an automated verification tool.**

***Index Terms*—Trust, Multi-Agent Systems, Parameter Adaptation, Validation, Verification.**

1 Introduction

When considering relations and interaction between agents, the concept of trust is of utmost importance. Within the domain of multi-agent systems, the concept of trust has been a topic of research for many years (e.g., Sabater and Sierra, 2005; Ramchurn et al., 2004). Within this research, the development of models expressing how agents form trust based upon direct experiences with a trustee or information obtained from parties other than the trustee is one of the central themes. Some of these models aim at creating trust models that can be utilized effectively within a software agent environment (e.g., van Maanen et al., 2007), whereas other models aim to present an accurate model of human trust (see e.g., Jonker and Treur, 1998; Falcone and Castelfranchi, 2004; Hoogendoorn et al., 2008). The latter type of model can be very useful when developing a personal assistant agent for a human with the awareness of the human's trust in different other agents (human or computer) and him- or herself (trustees). This could for example avoid advising to use particular information sources that are not trusted by the human or could be used to enhance the trust relationship with the personal assistant agent itself.

In order for computational trust models to be usable in real life settings, the validity of these models should be proven first. However, relatively few experiments have been performed that validate the accuracy of computational trust models upon empirical data. For instance, in (Jonker et al., 2004) an experiment has been conducted whereby the trends in human trust behavior have been analyzed to verify properties underlying trust models developed in the domain of multi-agent systems. However, no attempt was made to fit the model to the trusting behavior of the human.

In this paper, the results of a validation experiment for two computational trust models describing human trust are reported. An in previously studies utilized trust model (van Maanen et al., 2007), which was inspired on the trust model described in (Jonker and Treur, 1998), has been taken as a baseline model. This model uses experiences of per-

formance in order to estimate the trust in different trustees. The second model which is validated in this study is a model which also carries the notion of relative trust (Hoogendoorn et al., 2008). The idea of relative trust is that trust in a certain trustee not solely depends on the experiences with that trustee, but also with trustees that are considered competitors of that trustee. A comparison between the two models is also made to see whether the notion of relative trust describes human trust behavior in a more accurate way.

The validation process includes a number of steps. First, an experiment with participants has been performed in which trust plays an important role. As a result, empirical data has been obtained, that is usable for validating the two models. One part of the dataset is used to learn the best parameters for the two different trust models. Then these parameters are used to estimate human trust, using the same input as was used to generate the other part of the dataset. Finally, a number of assumptions underlying the two trust models are verified upon the obtained dataset using an automated verification tool.

This paper is organized as follows. First, the two trust models that have been used in this study are explained in Section 2. The experimental method is explained in Section 3. Thereafter, the results of the experiment in terms of model validation and verification are described in Section 4. Finally, Section 5 is a discussion.

2 Agent Models for Trust

In this section the two types of trust models which are subject of validation are described. In Section 2.1 a model is explained that estimates human trust in one trustee independent of the trust in other trustees. In contrast, in Section 2.2 a model is described for which this relative dependency actually is important.

2.1 Independent Trust Model

This section describes the independent trust model (van Maanen et al., 2007; Jonker and Treur, 1998). In this model trustees are considered rational and are therefore though of having no bias to calculate trust. Trust is based on experiences and there is a certain decay of trust.

For the present study, it is assumed that a set of trustees $\{S_1, S_2, \ldots, S_n\}$ is available that can be selected to give particular advice at each time step. Upon selection of one of the trustees (S_i), an experience is passed back indicating how well the trustee performed. This experience ($E_i(t)$) is a number on the interval $[-1, 1]$. Hereby, -1 expresses a negative experience, 0 is a neutral experience and 1 a positive experience. There is also a decay parameter λ_i in the model, for which holds that $0 \leq \lambda_i \leq 1$.

Given the above, trust now can calculated by means of the following formula:

$$T_i(t) = T_i(t-1) \cdot \lambda_i + (1 - \left(\frac{E_i(t)+1}{2}\right)) \cdot (1 - \lambda_i)$$

The independent trust is calculated for each trustee. Eventual reliance decisions are made by determining the maximum of the independent trust over all trustees.

2.2 Relative Trust Model

This section describes the relative trust model (Hoogendoorn et al., 2008). In this model trustees are considered competitors, and the human trust in a trustee depends on the relative experiences with the trustee to the experiences from the other trustees. The model defines the total trust of the human as the difference between positive trust and negative trust (distrust) on the trustee. The model includes several parameters representing human characteristics including trust flexibility β_i (measuring the change in trust on each new experience), decay γ_i (decay in trust when there is no experience) and autonomy η_i (dependence of the trust calculation considering other options). The model parameters β_i, γ_i and η_i have values from the interval $[0, 1]$.

As mentioned before, the model is composed of two models: one for positive trust, accumulating positive experiences, and one for negative trust, accumulating negative experiences. Both negative and positive trust are represented by a number between $[0, 1]$. The human's total trust $T_i(t)$ in S_i is the difference in positive and negative trust in S_i at time point t, which is a number between $[-1, 1]$, where -1 and 1 represent the minimum and maximum values of trust, respectively. The human's initial total trust in S_i at time point 0 is $T_i(0)$, which

is the difference in initial trust $T_i^+(0)$ and distrust $T_i^-(0)$ in S_i at time point 0.

As a differential equation the change in positive and negative trust over time is described in the following manner (Hoogendoorn et al., 2009b):

$$\frac{dT_i^+(t)}{dt} = E_i(t) \cdot \frac{(E_i(t)+1)}{2} \cdot \beta_i \cdot \Big(\eta_i \cdot (1 - T_i^+(t)) + (1-\eta_i) \cdot (\tau_i^+(t) - 1) \cdot T_i^+(t) \cdot (1 - T_i^+(t))\Big) - \gamma_i \cdot T_i^+(t) \cdot (1 + E_i(t)) \cdot (1 - E_i(t))$$

$$\frac{dT_i^-(t)}{dt} = E_i(t) \cdot \frac{(E_i(t)-1)}{2} \cdot \beta_i \cdot \Big(\eta_i \cdot (1 - T_i^-(t)) + (1-\eta_i) \cdot (\tau_i^-(t) - 1) \cdot T_i^-(t) \cdot (1 - T_i^-(t))\Big) - \gamma_i \cdot T_i^-(t) \cdot (1 + E_i(t)) \cdot (1 - E_i(t))$$

In these equations, $E_i(t)$ is the experience value given by S_i at time point t.

Furthermore, $\tau_i^+(t)$ and $\tau_i^-(t)$ are the human's relative positive and negative trust in S_i at time point t, which is the ratio of the human's positive or negative trust in S_i to the average human's positive or negative trust in all trustees at time point t defined as follows:

$$\tau_i^+(t) = \frac{T_i^+(t)}{\Big(\frac{\sum_{j=1}^n T_j^+(t)}{n}\Big)}$$

and

$$\tau_i^-(t) = \frac{T_i^-(t)}{\Big(\frac{\sum_{j=1}^n T_j^-(t)}{n}\Big)}$$

Finally, the total change in trust can be calculated as follows:

$$\frac{dT_i(t)}{dt} = \frac{dT_i^+(t)}{dt} - \frac{dT_i^-(t)}{dt}$$

Similarly as for the independent trust model, the trustee with the highest trust value is relied upon.

3 Method

In this section the experimental methodology is explained. In Section 3.1 the participants are described. In Section 3.2 an overview of the used experimental environment is given. Thereafter, the procedure of the experiment is explained in four stages: In Sections 3.3, 3.4, 3.5 and 3.6, the procedures of data collection, parameter adaptation, model validation and verification are explained, respectively. The results of the experiment are given in Section 4.

3.1 Participants

18 Participants (eight male and ten female) with an average age of 23 ($SD = 3.8$) participated in the experiment as paid volunteers. Participants were selected between the age of 20 and 30 and were not color blinded. All were experienced computer users, with an average of 16.2 hours of computer usage each week ($SD = 9.32$).

3.2 Task

The experimental task was a classification task in which two participants on two separate personal computers had to classify geographical areas according to specific criteria as areas that either needed to be attacked, helped or left alone by ground troops. The participants needed to base their classification on real-time computer generated video images that resembled video footage of real unmanned aerial vehicles (UAVs). On the camera images, multiple objects were shown. There were four kinds of objects: civilians, rebels, tanks and cars. The identification of the number of each of these object types was needed to perform the classification. Each object type had a score (either -2, -1, 0, 1 or 2, respectively) and the total score within an area had be determined. Based on this total score the participants could classify a geographical area (i.e., attack when above 2, help when below -2 or do nothing when in between). Participants had to classify two areas at the same time and in total 98 areas had to be classified. Both participants did the same areas with the same UAV video footage.

During the time a UAV flew over an area, three phases occurred: The first phase was the advice

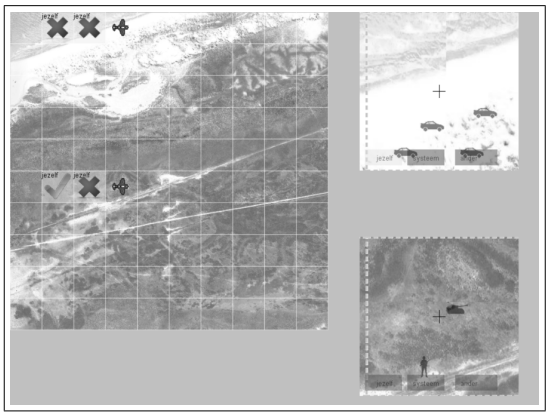


Figure 1. Interface of the task.

phase. In this phase both participants and a supporting software agent gave an advice about the proper classification (attack, help, or do nothing). This means that there were three advices at the end of this phase. It was also possible for the participants to refrain from giving an advice, but this hardly occurred. The second phase was the reliance phase. In this phase the advices of both the participants and that of the supporting software agent were communicated to each participant. Based on these advices the participants had to indicate which advice, and therefore which of the three trustees (self, other or software agent), they trusted the most. Participants were instructed to maximize the number of correct classifications at both phases (i.e., advice and reliance phase). The third phase was the feedback phase, in which the correct answer was given to both participants. Based on this feedback the participants could update their internal trust models for each trustee (self, other, software agent).

In Figure 1 the interface of the task is shown. The map is divided in 10×10 areas. These boxes are the areas that were classified. The first UAV starts in the top left corner and the second one left in the middle. The UAVs fly a predefined route so participants do not have to pay attention to navigation. The camera footage of the upper UAV is positioned top right and the other one bottom right.

The advice of the self, other and the software agent was communicated via dedicated boxes below the camera images. The advice to attack, help, or do nothing was communicated by red, green and yellow, respectively. On the overview screen on the left, feedback was communicated by the appearance of a green tick or a red cross. The reliance decision of the participant is also shown on the overview screen behind the feedback (feedback only shown in the feedback phase). The phase depicted in Figure 1 was the reliance phase before the participant indicated his reliance decision.

3.3 Data Collection

During the above described experiment, input and output were logged using a server-client application. The interface of this application is shown in Figure 2. Two other client machines, that were responsible for executing the task as described in the previous subsection, were able to connect via a local area network to the server, which was responsible for logging all data and communication between the clients. The interface shown in Figure 2 could be used to set the client's IP-addresses and ports, as well as several experimental settings, such as how to log the data.

Experienced performance feedback of each trustee and reliance decisions of each participant were logged in temporal order for later analysis. During the feedback phase the given feedback was translated to a penalty of either 0, $.5$ or 1, representing a good, neutral or poor experience of performance, respectively. This directly maps to the value $\frac{E_i(t)+1}{2}$ in the trust models. During the reliance phase the reliance decisions were translated to either 0 or 1 for each trustee S_i, which represented that one relied or did not rely on S_i.

3.4 Parameter Adaptation

The data collection described in Section 3.3 was repeated on each group of two participants twice, called condition 1 and condition 2, respectively. The data from one of the conditions was used for parameter adaptation purposes for both models, and the data from the other condition for model validation (see Section 3.5). This process of parameter adaptation and validation was balanced over conditions, which means that condition 1 and condition 2 switch roles (i.e., parameter adaptation and model validation) for half of the validation efforts (i.e., cross-validation). Both the parameter

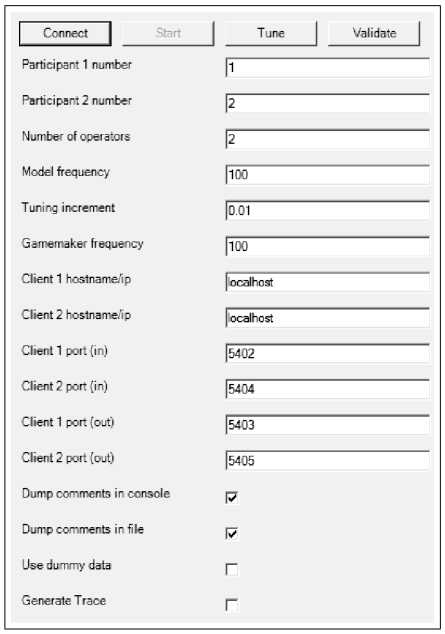


Figure 2. Interface of the application used for gathering validation data (Connect), for parameter adaptation (Tune) and validation of the trust models (Validate).

Algorithm 1 ES-PARAMETER-ADAPTATION(E, R_H)

1: $\delta_{\text{best}} \leftarrow \infty$
2: $X \leftarrow \mathbf{0}$
3: **for all** parameters x in parameter vector X **do**
4: **for all** settings of x **do**
5: $\delta_X \leftarrow 0$
6: **for all** time points t **do**
7: $e \leftarrow E(t)$
8: $r_M \leftarrow R_M(e, X)$
9: $r_H \leftarrow R_H(e)$
10: **if** $r_M \neq r_H$ **then**
11: $\delta_X \leftarrow \delta_X + 1$
12: **end if**
13: **end for**
14: **if** $\delta_X < \delta_{\text{best}}$ **then**
15: $X_{\text{best}} \leftarrow X$
16: $\delta_{\text{best}} \leftarrow \delta_X$
17: **end if**
18: **end for**
19: **end for**
20: **return** X_{best}

adaptation and model validation procedure was done using the same application as was used for gathering the empirical data. The interface shown in Figure 2 could also be used to alter validation and adaptation settings, such as the granularity of the adaptation.

The number of parameters of the models presented in Section 2 to be adapted for each model and each participant suggest that an exhaustive search (Hoogendoorn et al., 2009b) for the optimal parameters is feasible. This means that the entire parameter search space is explored to find a vector of parameter settings resulting in the maximum accuracy (i.e., the amount of overlap between the model's predicted reliance decisions and the actual human reliance decisions) for each of the models and each participant. The corresponding code of the implemented exhaustive search method is shown in Algorithm 1.

In this algorithm, $E(t)$ is the set of experiences (i.e., performance feedback) at time point t for all trustees, $R_H(e)$ is the actual reliance decision the participant made (on either one of the trustees) given a certain experience e, $R_M(e, X)$ is the predicted reliance decision of the trust model M (either independent or relative) given an experience e and candidate parameter vector X (reliance on either one of the trustees), δ_X is the distance between the estimated and actual reliance decisions given a certain candidate parameter vector X, δ_{best} is the distance resulting from the best parameter vector X_{best} found so far. The best parameter vector X_{best} is returned when the algorithm finishes. This parameter adaptation procedure was implemented in C#. Part of the C#-code is listed in the appendix of this paper, where the method "UpdateDistance()" corresponds to lines 5 until 10 in Algorithm 1 and $R_M(e, X)$ is calculated by the method "TrusteeWithMaxTrust()".

If for Algorithm 1 the number of parameters is μ, Γ the granularity for each parameter, N the number of trustees and B the number of reliance decisions (i.e., time points) made by the human, then the worst case complexity of the algorithm is expressed as $O(10^{\mu\Gamma}BN)$. The complexity also depends on N, since $R_M(e, X)$ results in a calculation of trust

values over all trustees. For the independent trust model it holds that $\mu = 1$ (i.e., the parameter λ_i) and for the relative trust model $\mu = 3$ (i.e., the three parameters β_i, γ_i and η_i). In the current experiment it furthermore holds that $\Gamma = 2$ (i.e., steps of .01), $N = 3$ (the two humans and the software agent) and $B = 98$ (the total of classified geographical areas). This means that $2.94 \cdot 10^4$ computation steps are needed for the independent trust model and $2.94 \cdot 10^8$ for the relative trust model, which took on average 31 milliseconds for the first, and 3 minutes and 20 seconds computation time for the second model.[1]

3.5 Validation

In order to validate the two models described in Section 2, the measurements of experienced performance feedback were used as input for the models and the output (predicted reliance decisions) of the models was compared with the the actual reliance decisions of the participant. The overlap of the predicted and the actual reliance decisions was a measure for the accuracy of the models. The results are in the form of dynamic accuracies over time, average accuracy per condition (1 or 2) and per trust model (independent or relative). A comparison between the averages per model and the interaction effect between condition role allocation (i.e., parameter adaptation either in condition 1 or 2) and model type, is done using a repeated measures analysis of variance (ANOVA).

3.6 Verification

Next to a validation using the accuracy of the prediction using the models, another approach has been used to validate the assumptions underlying existing trust models. The idea is that properties that form the basis of trust models are verified against the empirical results obtained within the experiment. In order to conduct such an automated verification, the properties have been specified in a language called Temporal Trace Language (TTL) (Bosse et al., 2009) that features a dedicated editor and an automated checker. The language TTL is explained first, followed by an expression of the desired properties related to trust.

Temporal Trace Language (TTL): The predicate logical temporal language TTL supports formal specification and analysis of dynamic properties, covering both qualitative and quantitative aspects. TTL is built on atoms referring to states of the world, time points and traces, i.e., trajectories of states over time. In addition, dynamic properties are temporal statements that can be formulated with respect to traces based on the state ontology Ont in the following manner. Given a trace γ over state ontology Ont, the state in γ at time point t is denoted by state(γ, t). These states can be related to state properties via the formally defined satisfaction relation denoted by the infix predicate $\models$, i.e., state(γ, t) $\models p$ denotes that state property p holds in trace γ at time t. Based on these statements, dynamic properties can be formulated in a formal manner in a sorted first-order predicate logic, using quantifiers over time and traces and the usual first-order logical connectives such as $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\forall$ and $\exists$. For more details on TTL, see (Bosse et al., 2009).

Properties for Trust Models: Within the literature on trust, a variety of properties have been expressed concerning the desired behavior of trust models. In many of these properties, the trust values are explicitly referred to, for instance in the work of (Jonker and Treur, 1998) characteristics of trust models have been defined (e.g., monotonicity and positive trust extension upon positive experiences). In this paper however, the trust function is subject of validation and therefore cannot be taken as a basis. Therefore, properties are expressed on an external basis, solely using the information which has been observed within the experiment to see whether these behaviors indeed comply to the desired behavior of the trust models. This information is then limited to the experiences that are received as an input and the choices that are made by the human that are generated as output. The properties from (Hoogendoorn et al., 2009a) are taken as a basis for these properties. Essentially, the properties indicate the following desired behavior of human trust:

1) Positive experiences lead to higher trust
2) Negative experiences lead to lower trust
3) Most trusted trustee is selected

[1] This was on an ordinary PC with an Intel(R) Core(TM)2 Quad CPU @2.40 GHz inside. Note that $31 \cdot \frac{2.94 \cdot 10^8}{2.94 \cdot 10^4}$ milliseconds $=$ 5.17 minutes $\neq$ 3.33 minutes computation time. This is due to a fixed initialization time of on average 11 ms for both models.

As can be seen, the properties also use the intermediate state of trust. In order to avoid this, it is however possible to combine these properties into a single property that expresses a relation between the experiences and the selection (i.e., the above items $1 + 3$ and $2 + 3$). Two of these properties are shown below. In addition, a property is expressed which specifies the notion of relativity in the experiences and the resulting selection of a trustee. The first property expresses that a trustee that gives the absolute best experiences during a certain period is eventually selected at least once within, or just after that particular period, and is shown below.

P1(min_duration, max_duration, max_time): Absolute more positive experiences results in selection
If a trustee a_1 always gives more positive experiences than all other trustees during a certain period with minimal duration min_duration and maximum duration max_duration, then this trustee a_1 is selected at least once during the period [min_duration, max_duration + max_time].

Formal:

P1(min_duration:DURATION, max_duration:DURATION, max_delay:DURATION) ≡
$\forall\gamma$:TRACE, $t_{\text{start}}, t_{\text{end}}$:TIME, a:TRUSTEE
[[$t_{\text{end}} - t_{\text{start}} \geq$ min_duration & $t_{\text{end}} - t_{\text{start}} \leq$ max_duration & absolute_highest_experiences(γ, a, t_{start}, t_{end})
$\Rightarrow$ selected(γ, a, t_{start}, t_{end}, max_delay)

where

absolute_highest_experiences(γ:TRACE, a:TRUSTEE, t_{start}:TIME, t_{end}:TIME) ≡
$\forall t$:TIME, r_1, r_2:REAL, a_2:TRUSTEE $\neq a$
[[$t \geq t_{\text{start}}$ & t $< t_{\text{end}}$ &
state(γ, t) $\models$ trustee_gives_experience(a, r_1) &
state(γ, t) $\models$ trustee_gives_experience(a_2, r_2)]
$\Rightarrow r_2 < r_1$]

selected(γ:TRACE, a:TRUSTEE, t_{start}:TIME, t_{end}:TIME, z:DURATION) ≡
$\exists t$:TIME [$t \geq t_{\text{start}}$ & $t < t_{\text{end}} + z$ &
state(γ, t) $\models$ trustee_selected(a)]

The second property, P2, specifies that the trustee which gives more positive experiences on average during a certain period is at least selected once within or just after that period.

P2(min_duration, max_duration, max_time, higher_exp): Average more positive experiences results in selection
If a trustee a_1 on average gives the most positive experiences (on average more than higher_exp better than the second best) during a period with minimal duration min_duration and maximum duration max_duration, then this trustee a_1 is selected at least once during the period [min_duration, max_duration+max_time].

Formal:

P2(min_duration:DURATION, max_duration:DURATION, max_delay:DURATION, higher_exp:REAL) ≡
$\forall\gamma$:TRACE, $t_{\text{start}}, t_{\text{end}}$:TIME, a:TRUSTEE
[[$t_{\text{end}} - t_{\text{start}} \geq$ min_duration & $t_{\text{end}} - t_{\text{start}} \leq$ max_duration & average_highest_experiences(γ, a, t_{start}, t_{end}, higher_exp)]
$\Rightarrow$ selected(γ, a, t_{start}, t_{end}, max_delay)]

where

average_highest_experiences(γ:TRACE, a:TRUSTEE, t_{start}:TIME, t_{end}:TIME, higher_exp:REAL) ≡
$\forall t$:TIME, r_1, r_2:REAL, a_2:TRUSTEE $\neq a$
[$t \geq t_{\text{start}}$ & $t < t_{\text{end}}$ &
[$\sum_{\forall t:\text{TIME}}$ case(experience_received(γ, a, t, t_{start}, t_{end}, e), e, 0) >
($\sum_{\forall t:\text{TIME}}$ (case(experience_received(γ, a, t, t_{start}, t_{end}, e), e, 0)) + higher_exp * $t_{\text{end}} - t_{\text{start}}$)]]

In the formula above, the case(p, e, 0) operator evaluates to e in case property p is satisfied and to 0 otherwise.

experience_received(γ:TRACE, a:TRUSTEE, t:TIME, t_{start}:TIME, t_{end}:TIME, r:REAL) ≡
[$\exists r$:REAL, $t \geq t_{\text{start}}$ & $t < t_{\text{end}}$ &
state(γ, t) $\models$ trustee_gives_experience(a, r)]

The final property concerns the notion of relativity which plays a key role in the models verified throughout this paper. The property expresses that the frequency of selection of a trustee that gives an identical experience pattern during two periods is not identical in case the other trustees give different experiences.

P3(interval_length, min_difference, max_time): Relative trust
If a trustee a_1 gives an identical experience pattern during two periods [t_1, t_1+ interval_length] and [t_2, t_2+ interval_length] and the experiences of at least one other trustee is not identical (i.e., more than min_difference different at each time point), then the selection frequency of a_1 will be different in a period during, or just after the specified interval.

Formal:

P3(interval_length:DURATION, min_difference:REAL, max_time:DURATION) $\equiv$
$\forall\gamma$:TRACE, t_1, t_2:TIME, a:TRUSTEE
[[same_experience_sequence(γ, a, t_1, t_2, interval_length) & $\exists a_2$:TRUSTEE $\neq a$
[different_experience_sequence(γ, a, t_1, t_2, min_difference)]
$\Rightarrow \exists i$:DURATION $<$ max_time
$\sum_{\forall t:\text{TIME}}$ case(selected_option(γ, a, t, $t_1 + i$, $t_1 + i$+ interval_length), 1, 0) /
(1 + $\sum_{\forall t:\text{TIME}}$ case(trustee_selected(γ, t, t_1, $t_1 + i$+ interval_length), 1, 0)) $\neq$
$\sum_{\forall t:\text{TIME}}$ case(selected_option(γ, a, t, $t_2 + i$, $t_2 + i$+ interval_length), 1, 0) /
(1 + $\sum_{\forall t:\text{TIME}}$ case(trustee_selected(γ, t, $t_2 + i$, $t_2 + i$+ interval_length), 1, 0))

where

same_experience_sequence(γ:TRACE, a:TRUSTEE, t_1:TIME, t_2:TIME, x:DURATION) $\equiv$
$\forall y$:DURATION [$y \geq 0$ & $y \leq x$ & $\exists r$:REAL
[state(γ, $t_1 + y$) $\models$ trustee_gives_experience(a, r) &
state(γ, $t_2 + y$) $\models$ trustee_gives_experience(a, r)]]

different_experience_sequence(γ:TRACE, a:TRUSTEE, t_1:TIME, t_2:TIME, x:DURATION, min_difference:REAL) $\equiv$
$\forall y$:DURATION [$y \geq 0$ & $y \leq x$ & $\exists r_1$, r_2:REAL
[state(γ, $t_1 + y$) $\models$ trustee_gives_experience(a, r_1) &
state(γ, $t_2 + y$) $\models$ trustee_gives_experience(a, r_2) &
$|r_1 - r_2| >$ min_difference]]

trustee_selected(γ:TRACE, t:TIME, t_{start}:TIME, t_{end}:TIME) $\equiv \exists a$:TRUSTEE
[$t \geq t_{\text{start}}$ & $t < t_{\text{end}}$ & state(γ, t) $\models$ trustee_selected(a)]

4 Results

In this section the validation and verification results are given in Sections 4.1 and 4.2, respectively.

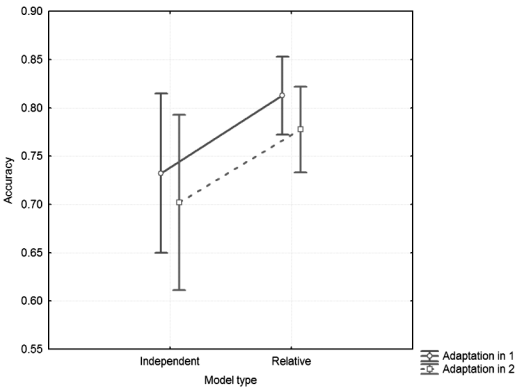


Figure 4. Interaction effect between condition role allocation and model type on accuracy.

4.1 Validation Results

From the data of 18 participants, one dataset has been removed due to an error while gathering data. This means that there are 2 (condition role allocations, i.e., parameter adaptation either in condition 1 or 2) times 17 (participants) $=$ 34 data pairs (accuracies for 2 models). Due to a significant Grubbs test, from these pairs 3 outliers were removed. Hence in total 31 pairs were used for the data analysis.

In Figure 3 the main effect of model type (either independent or relative trust) for accuracy is shown. A repeated measures analysis of variance (ANOVA) showed a significant main effect ($F(1, 29) = 7.60$, $p < .01$). This means that indeed the relative trust model had a higher accuracy ($M = .7968$, $SD = .0819$) than the independent trust model ($M = .7185$, $SD = .1642$).

Figure 4 shows the possible interaction effect between condition role allocation (parameter adaptation in condition 1 or in condition 2) and model type (either independent or relative trust) on accuracy. No significant interaction effect was found ($F(1, 29) = 0.01$, $p = .93$). Hence no significant learning effect between conditions was found. Cross-validation was not needed to balance the data, but the procedure still produced twice as much data pairs.

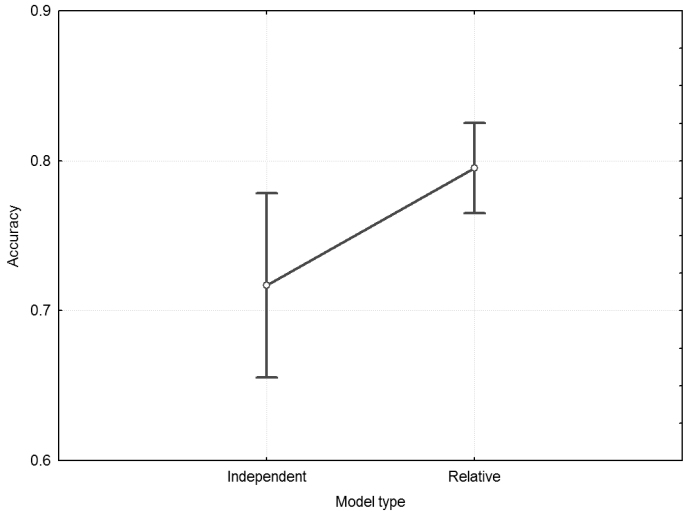


Figure 3. Main effect of model type for accuracy.

Table I
RESULTS OF VERIFICATION OF PROPERTY P1 AND P2.

min_duration	% satisfying P1	% satisfying P2
1	64.7	29.4
2	64.7	29.4
3	86.7	52.9
4	92.3	55.9
5	100.0	58.8
6	100.0	70.6

4.2 Verification Results

The results of the verification of the properties against the empirical traces (i.e., formalized logs of human behavior observed during the experiment) are shown in Table I. First, the results for properties P1 and P2 are shown. Hereby, the value of max_duration has been kept constant at 30 and the max_time after which the trustee should be consulted is set to 5. The minimal interval time (min_duration) has been varied. Finally, for property P2 the variable higher_exp indicating how much higher the experience should be on average compared to the other trustees is set to .5.

The results in Table I indicate the percentage of traces in which the property holds out of all traces in which the antecedent at least holds once (i.e., at least one sequence with the min_duration occurs in the trace). This has been done to avoid a high percentage of satisfaction due to the fact that in some of the traces the antecedent never holds, and hence, the property is always satisfied. The table shows that the percentage of traces satisfying P1 goes up as the minimum duration of the interval during which a trustee gives the highest experience increases. This clearly complies to the ideas underlying trust models as the longer a trustee gives the highest experiences, the higher his trust will be (also compared to the other trustees), and the more likely it is that the trustee will be selected. The second property, counting the average experience and its implication upon the selection behavior of the human, also shows an increasing trend in satisfaction of the property with the duration of the interval during which the trustee on average gives better experiences. The percentages are lower compared to P1 which can be explained by the fact that they might also give some negative experiences compared to the alternatives (whereas they are giving better experiences on average). This could then result in a decrease in the trust value, and hence, a lower probability of being selected.

The third property, regarding the relativity of

Table II
RESULTS OF VERIFICATION OF PROPERTY P3.

interval_length	% satisfying P3
1	0
2	41.1
3	55.9
4	67.6
5	66.7
6	68.4

trust has also been verified and the results of this verification are shown in Table II. Here, the traces of the participants have been verified with a setting of min_difference to .5 and max_time to 5 and the variable interval_length during which at least one trustee shows identical experiences whereas another shows different experiences has been varied.

It can be seen that property P3 holds more frequently as the length of the interval increases, which makes sense as the human has more time to perceive the relative difference between the two. Hence, this shows that the notion of relative trust can be seen in the human trustee selection behavior in almost 70% of the cases.

5 DISCUSSION AND CONCLUSIONS

In this paper, an extensive validation study has been performed to show that human trust behavior can be accurately described and predicted using computational trust models. In order to do so, an experiment has been designed that places humans in a setting where they have to make decisions based upon the trust they have in others. In total 18 participants took part in the experiment. The results show that both an independent (see van Maanen et al., 2007; Jonker and Treur, 1998) as well as a relative trust model (see Hoogendoorn et al., 2008) can predict this behavior with a high accuracy (72% and 80%, respectively) by learning on one dataset and predicting the trust behavior for another (cross-validation). Furthermore, it has also been shown that the underlying assumptions of the trust models (and many other trust models) are found in the data of the participants.

Of course, more work on the validation of trust models has been performed. In (Jonker et al., 2004) an experiment has been presented in which human experiments in trust have been described. Although the underlying assumptions of trust models have to some extent been verified in that paper, no attempt has been made to fit a trust model to the data. Other papers describing the validation of trust models for instance validate the accuracy of trust models describing the propagation of trust through a network (e.g., Guha et al., 2004). In (McKnight et al., 2001) a multidisciplinary and multidimensional model of trust in e-commerce is validated. The model includes four high-level constructs: 1) disposition to trust, 2) institution-based trust, 3) trusting beliefs and 4) trusting intentions. The proposed model itself does however not describe the formation of trust on such a detailed level as the models used throughout this paper, it presents general relationships between trust measures and these relationships are subject to validation. Gefen and Straub (2004) validate a four-dimensional scale of trust in the context of e-Products and revalidates it in the context of e-Services which shows the influence of social presence on these dimensions of trust, especially benevolence, and its ultimate contribution to online purchase intentions. Again, correlations are found between the concepts of trust that have been distinguished, but no computational model for the formation of trust and the precise prediction thereof is prosed. Finally, in (da Costa Hernandez and dos Santos, 2010) a development-based trust measurement model for buyer-seller relationships is presented and validated against a characteristic-based trust measurement model in terms of its ability to explain certain variables of interest in buyer-seller relationships (long-term relationship orientation, information sharing, behavioral loyalty and future intentions).

Within the domain of agent systems, quite some trust models have been developed (for an overview, see e.g., Sabater and Sierra, 2005; Ramchurn et al., 2004). Although the focus of this paper has been on the validation of two specific trust models, thereby also comparing relative with absolute trust, other trust models can also be validated using the experimental data obtained in combination with parameter estimation. This is part of the future work. Furthermore, other parameter adaptation methods will be explored or extended for the purpose of real-time adaptation, which accounts for human learning. In addition, a personal assistant software agent will

be implemented that is able to monitor and balance the functional state of the human in a timely and knowledgeable manner. Also applications in different domains are explorable, such as the military and air traffic control domain.

ACKNOWLEDGMENTS

This research was partly funded by the Dutch Ministry of Defense under progr. no. V929. Furthermore, this research has partly been conducted as part of the FP7 ICT Future Enabling Technologies program of the European Commission under grant agreement no. 231288 (SOCIONICAL). The authors would like to acknowledge Francien Wisse for her efforts to gather the necessary validation data and implementing the experimental task. The authors would also like to thank Tibor Bosse, Jan-Willem Streefkerk and Jan Treur for their helpful comments.

REFERENCES

Bosse, T., Jonker, C. M., van der Meij, L., Sharpanskykh, A., and Treur, J. (2009). Specification and verification of dynamics in agent models. *International Journal of Cooperative Information Systems*, 18:167–193.

da Costa Hernandez, J. M. and dos Santos, C. C. (2010). Development-based trust: Proposing and validating a new trust measurement model for buyer-seller relationships. *Brazilian Administration Review*, 7:172–197.

Falcone, R. and Castelfranchi, C. (2004). Trust dynamics: How trust is influenced by direct experiences and by trust itself. In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04)*, pages 740–747, New York, USA.

Gefen, D. and Straub, D. W. (2004). Consumer trust in b2c e-commerce and the importance of social presence: experiments in e-products and e-services. *Omega*, 32:407–424.

Guha, R., Kumar, R., Raghavan, P., and Tomkins, A. (2004). Propagation of trust and distrust. In *Proceedings of the 13th international conference on World Wide Web (WWW '04)*, pages 403–412, New York, NY. ACM.

Hoogendoorn, M., Jaffry, S., and Treur, J. (2008). Modeling dynamics of relative trust of competitive information agents. In Klusch, M., Pechoucek, M., and Polleres, A., editors, *Proceedings of the 12th International Workshop on Cooperative Information Agents (CIA'08)*, volume 5180 of *LNAI*, pages 55–70. Springer.

Hoogendoorn, M., Jaffry, S., and Treur, J. (2009a). Modelling trust dynamics from a neurological perspective. In *Proceedings of the Second International Conference on Cognitive Neurodynamics (ICCN'09)*. Springer Verlag. To appear.

Hoogendoorn, M., Jaffry, S. W., and Treur, J. (2009b). An adaptive agent model estimating human trust in information sources. In Baeza-Yates, R., Lang, J., Mitra, S., Parsons, S., and Pasi, G., editors, *Proceedings of the 9th IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'09)*, pages 458–465.

Jonker, C. M., Schalken, J. J. P., Theeuwes, J., and Treur, J. (2004). Human experiments in trust dynamics. In *Proceedings of the Second International Conference on Trust Management (iTrust 2004)*, volume 2995 of *LNCS*, pages 206–220. Springer Verlag.

Jonker, C. M. and Treur, J. (1998). Formal analysis of models for the dynamics of trust based on experiences. In Garijo, F. J. and Boman, M., editors, *Multi-Agent System Engineering, Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW'99*, volume 1647, pages 221–232, Berlin. Springer Verlag.

McKnight, D. H., Choudhury, V., and Kacmar, C. (2001). Developing and validatin trust measures for e-commerce: An integrative topology. *Information Systems Research*, 13(3):334–359.

Ramchurn, S. D., Huynh, D., and Jennings, N. R. (2004). Trust in multi-agent systems. *The Knowledge Engineering Review*, 19(1):1–25.

Sabater, J. and Sierra, C. (2005). Review on computational trust and reputation models. *Artificial Intelligence Review*, 24(1):33–60.

van Maanen, P.-P., Klos, T., and van Dongen, K. (2007). Aiding human reliance decision making using computational models of trust. In *Proceedings of the Workshop on Communication between Human and Artificial Agents (CHAA'07)*, pages 372–376, Fremont, California, USA. IEEE Computer Society Press. Co-located with The

2007 IEEE IAT/WIC/ACM International Conference on Intelligent Agent Technology.

APPENDIX

In this appendix part of the C#-code that was used for parameter adaptation and model validation is listed and described. This part is called the DModel class and is used to calculate the trust values at a certain time point (that is, for one of the areas) and given a certain operator (either of the two operators), certain parameter settings (within certain intervals) and model type (either the independent or relative trust model). The in this appendix listed code consists of the following four methods:

1) **Dmodel:** This is the constructor of the DModel class, which initializes the independent or relative trust model. For each operator (indicated by "operatornumber") and each model type (indicated by "type") a DModel-object is created.
2) **UpdateValuesIndepdendentModel:** Calculates the independent trust value (called "data.values") for each trustee (the operator/-participant him- or herself, the other participant and the supporting software agent).
3) **UpdateValuesRelativeModel:** Calculates the relative trust value for each trustee. The "UpdateValues"-methods are called from the parent class of DModel for each time step.[2]
4) **UpdateDistance:** Calculates the distance (depicted by δ_X in Algorithm 1) between the by the trust model predicted reliance decision (depicted by $R_M(e, X)$ in Algorithm 1) and the actual human reliance decision (depicted by $R_H(e)$ in Algorithm 1). For parameter adaptation purposes, this method is called from the parent class of DModel for each time step. This is done after either "UpdateValuesIndepdendentModel" or "UpdateValuesRelativeModel" is called to calculate the current trust values. In this procedure values for "data.dmodelParameters" are altered as is shown in Algorithm 1.
5) **TrusteeWithMaxTrust:** Calculates the trustee for which there is currently the maximum trust value. This method is called from the method "UpdateDistance()" in order to determine the predicted reliance decision. The current trust values are calculated either by the above described "UpdateValuesIndepdendentModel()" or the "UpdateValuesRelativeModel()" method.

[2]Due to limitations of space, the code of the parent class of DModel is omitted. Those further interested in this code are referred to `http://www.few.vu.nl/~pp/trust`.

Below the code of the DModel class is listed.

```
namespace UAVtrustServer
{
  /// <summary>
  /// This class was written by Waqar Jaffry
      and Peter-Paul van Maanen 2010.
     Nothing of this code may be used or
     copied without the permission of the
     authors. This software estimates the
     current trust of a UAV-operator (
     operator 1 or 2) in different trustees
      (self, other and system).
  /// </summary>

  public class DModel : Model {
    double[] PositiveTrust;
    double[] NegativeTrust;

    public DModel(int opnr, Data d, bool
        tune)
      : base(0, opnr, d, tune) {
      // Constructor partly inherited from
          parent (code ommited)
      PositiveTrust = new double[data.values
          .GetLength(2)];
      NegativeTrust = new double[data.values
          .GetLength(2)];
    }

    // Method called to do iteration of the
        independent trust model to update to
       the next value
    public override void
        UpdateValuesIndependentModel() {
      int gridx, gridy;
      double totalpenalty, newtrust,
          oldtrust;
      double decay = data.dmodelParameters[
          operatornumber, 0];

      // Update trust for each trustee
      for (int trusteenr = 0; trusteenr <
          data.values.GetLength(2);
          trusteenr++) {
        // Use default values or values from
            the parameters file
        if (data.modelLoopNumber[
            operatornumber, type] == -1)
          data.values[operatornumber, type,
              trusteenr] = data.
              dmodelParameters[
              operatornumber, trusteenr +
```

```
                1];
29          else { // Otherwise calculate the
                new trust values
              totalpenalty = 0;
31
              // Update trust for each UAV
33            for (int uavnr = 0; uavnr < data.
                  currentUavWorld[operatornumber
                  , type].uavs.Length; uavnr++)
                  {
                gridx = data.currentUavWorld[
                    operatornumber, type].
                    lastfeedback[uavnr, 0];
35              gridy = data.currentUavWorld[
                    operatornumber, type].
                    lastfeedback[uavnr, 1];

37              totalpenalty += data.
                    currentUavWorld[
                    operatornumber, type].grid[
                    gridx, gridy].penalty[0,
                    trusteenr];
              }
39
              newtrust = totalpenalty / data.
                  currentUavWorld[operatornumber
                  , type].uavs.Length;
41            oldtrust = data.values[
                  operatornumber, type,
                  trusteenr];
              data.values[operatornumber, type,
                  trusteenr] = decay * oldtrust
                  + (1 - decay) * newtrust;
43            data.dmodelParameters[
                  operatornumber, 1 + trusteenr]
                   = data.values[operatornumber,
                   type, trusteenr]; // For
                  storing the last value as
                  parameter
            }
45        }
      }
47
      // Method called to do iteration of the
          relative trust model to update to
          the next value
49    public override void
          UpdateValuesRelativeModel() {
        double[] deltaPositiveTrust;
51      double[] deltaNegativeTrust;
        double SigmaPositiveTrust = 0,
            SigmaNegativeTrust = 0;
53      deltaNegativeTrust = new double[data.
            values.GetLength(2)];
        deltaPositiveTrust = new double[data.
            values.GetLength(2)];
55      int gridx, gridy;
        double Gama = data.dmodelParameters[
            operatornumber, 0];
57          double Beta = data.
                dmodelParameters[
                operatornumber, 1];
            double Eta = data.
                dmodelParameters[
                operatornumber, 2];
59      double INTERVAL_LENGTH = 0.1;

61      // Use default values or values from
            the parameters file (after
            adaptation)
          if (data.modelLoopNumber[
              operatornumber, type] == -1)
              {
63        PositiveTrust[0] = data.
              dmodelParameters[operatornumber,
               3];
          NegativeTrust[0] = data.
              dmodelParameters[operatornumber,
               4];
65        PositiveTrust[1] = data.
              dmodelParameters[operatornumber,
               5];
          NegativeTrust[1] = data.
              dmodelParameters[operatornumber,
               6];
67        PositiveTrust[2] = data.
              dmodelParameters[operatornumber,
               7];
          NegativeTrust[2] = data.
              dmodelParameters[operatornumber,
               8];
69      }
        else {
71        // Update trust value for each UAV
          for (int uavnr = 0; uavnr < data.
              currentUavWorld[operatornumber,
              type].uavs.Length; uavnr++) {
73          double penalty = 0;
            gridx = data.currentUavWorld[
                operatornumber, type].
                lastfeedback[uavnr, 0];
75          gridy = data.currentUavWorld[
                operatornumber, type].
                lastfeedback[uavnr, 1];

77          SigmaPositiveTrust = 0;
            SigmaNegativeTrust = 0;
79          for (int truseern = 0; truseern <
                data.values.GetLength(2);
                truseern += 1) {
              SigmaPositiveTrust +=
                  PositiveTrust[truseern];
81            SigmaNegativeTrust +=
                  NegativeTrust[truseern];
            }
83
            // Update trust value for each
                trustee
85          for (int CurrentTrustee = 0;
                CurrentTrustee < data.values.
                GetLength(2); CurrentTrustee
                += 1) {
              penalty = data.currentUavWorld[
                  operatornumber, type].grid[
                  gridx, gridy].penalty[0,
```

```
                CurrentTrustee];

            if (penalty < 0.5) {
              deltaPositiveTrust[
                  CurrentTrustee] = Beta * (
                  Eta * (1 - PositiveTrust[
                  CurrentTrustee]) - (1 -
                  Eta) * (1 - data.values.
                  GetLength(2) *
                  PositiveTrust[
                  CurrentTrustee] / (
                  SigmaPositiveTrust)) *
                  PositiveTrust[
                  CurrentTrustee] * (1 -
                  PositiveTrust[
                  CurrentTrustee])) *
                  INTERVAL_LENGTH;
              deltaNegativeTrust[
                  CurrentTrustee] = (-(1 -
                  Gama) * NegativeTrust[
                  CurrentTrustee] *
                  INTERVAL_LENGTH);
            }
            else if (penalty == 0.5) {
              deltaPositiveTrust[
                  CurrentTrustee] = (-(1 -
                  Gama) * PositiveTrust[
                  CurrentTrustee] *
                  INTERVAL_LENGTH);
              deltaNegativeTrust[
                  CurrentTrustee] = (-(1 -
                  Gama) * NegativeTrust[
                  CurrentTrustee] *
                  INTERVAL_LENGTH);
            }
            else {
              deltaNegativeTrust[
                  CurrentTrustee] = Beta * (
                  Eta * (1 - NegativeTrust[
                  CurrentTrustee]) - (1 -
                  Eta) * (1 - data.values.
                  GetLength(2) *
                  NegativeTrust[
                  CurrentTrustee] / (
                  SigmaNegativeTrust)) *
                  NegativeTrust[
                  CurrentTrustee] * (1 -
                  NegativeTrust[
                  CurrentTrustee])) *
                  INTERVAL_LENGTH;
              deltaPositiveTrust[
                  CurrentTrustee] = (-(1 -
                  Gama) * PositiveTrust[
                  CurrentTrustee] *
                  INTERVAL_LENGTH);
            }
          }

          for (int CurrentTrustee = 0;
              CurrentTrustee < data.values.
              GetLength(2); CurrentTrustee
              += 1) {
            PositiveTrust[CurrentTrustee] +=
                deltaPositiveTrust[
                CurrentTrustee];
            NegativeTrust[CurrentTrustee] +=
                deltaNegativeTrust[
                CurrentTrustee];
          }
        }
      }

      for (int CurrentTrustee = 0;
          CurrentTrustee < data.values.
          GetLength(2); CurrentTrustee++) {
        data.values[operatornumber, type,
            CurrentTrustee] = (PositiveTrust
            [CurrentTrustee] - NegativeTrust
            [CurrentTrustee] + 1) / 2;
        data.dmodelParameters[operatornumber
            , CurrentTrustee * 2 + 3] =
            PositiveTrust[CurrentTrustee];
            // For storing the last value as
             parameter
        data.dmodelParameters[operatornumber
            , CurrentTrustee * 2 + 4] =
            NegativeTrust[CurrentTrustee];
            // For storing the last value as
             parameter
      }
    }

    // Update the distance between the
        generated model output (either
        independent or relative) and the
        validation data for one time step
    // The trust values (data.values) have
        already been updated for the 3
        trustees given the current parameter
         settings (data.dmodelParameters)
    public override void UpdateDistance() {
      int gridx, gridy, distance, maxtrust;

      // Update the distance to the proper
          value
      for (int uavnr = 0; uavnr < data.
          currentUavWorld[operatornumber,
          type].uavs.GetLength(0); uavnr++)
          {
        gridx = data.currentUavWorld[
            operatornumber, type].
            lastfeedback[uavnr, 0];
        gridy = data.currentUavWorld[
            operatornumber, type].
            lastfeedback[uavnr, 1];

        maxtrust = TrusteeWithMaxTrust();

        // distance == 0 when indeed the
            human relied on the trustee with
             the highest model's estimated
            trust value, otherwise the
            distance is higher (1 or 2)
        if (data.currentUavWorld[
            operatornumber, type].grid[gridx
```

```
            , gridy].reliedon[0, maxtrust]
            == 1)
          distance = 0;
131       else if (data.currentUavWorld[
            operatornumber, type].grid[gridx
            , gridy].reliedon[0, 0] != 1 &&
            data.currentUavWorld[
            operatornumber, type].grid[gridx
            , gridy].reliedon[0, 1] != 1 &&
            data.currentUavWorld[
            operatornumber, type].grid[gridx
            , gridy].reliedon[0, 2] != 1)
          distance = 0;
133       else
          distance = 1;
135
          // Update value according to the
             given distance
137       data.modelDistance[operatornumber,
            type] += distance;
        }
139   }

141   // Return the trustee (0, 1 or 2) with
        the most trust according to the
        current trust values (data.values)
      public int TrusteeWithMaxTrust() {
143     int index = 0;
        for (int trusteenr = 1; trusteenr <
            data.values.GetLength(2);
            trusteenr++) // for all trustees
145       if (data.values[operatornumber, type
            , trusteenr] > data.values[
            operatornumber, type, index])
          index = trusteenr;
147
        return index;
149   }
    }
151 }
```

dmodel.cs

Chapter 8

Effects of Reliance Support on Team Performance by Advising and Adaptive Autonomy

Effects of Reliance Support on Team Performance by Advising and Adaptive Autonomy

Peter-Paul van Maanen*†, Francien Wisse‡, Jurriaan van Diggelen* and Robbert-Jan Beun‡

* Department of Cognitive Systems Engineering, TNO Human Factors
P.O. Box 23, 3769 ZG Soesterberg, The Netherlands
Email: {peter-paul.vanmaanen, jurriaan.vandiggelen}@tno.nl
† Department of Artificial Intelligence, Vrije Universiteit Amsterdam
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
‡ Department of Information and Computing Sciences, Utrecht University
P.O. Box 80089, 3508 TB Utrecht, The Netherlands
Email: rj@cs.uu.nl

***Abstract*—Problems with estimating trust in information sources are common in time constraining and ambiguous situations and often lead to a decrease of team performance. Humans lack the resources to track the integrity of information and thus tend to over- or under-rely on advice from support systems. Two types of adaptive team support have been developed and evaluated: The first adaptive system (graphical support) supports by communicating the estimated degree of over- or under-trust. The second system (adaptive autonomy) takes over a reliance decision when this estimation exceeds a certain threshold. The two types of support were implemented in a multi-agent environment where human operators and Unmanned Aerial Vehicles (UAVs) work together on a target classification task. An experiment is reported in which the above two support types were evaluated in that multi-agent environment in terms of team performance, satisfaction and effectiveness due to competence and task difficulty. Team performance in the support conditions were somewhat higher compared to no support. However, these differences were not significant. A significant increased effect was found for participants that performed less well. The results also show significantly less satisfaction when applying adaptive autonomy compared to advising through the graphical support.**

1 Introduction

In many domains such as aviation, military, air traffic control and crisis management, decisions are more and more based on advice of decision support systems. This is inevitable because of reduction of staff, their increased responsibilities and the increasing complexity of the tasks (Grootjen and Neerincx, 2005).

Many studies emphasize the importance of *trust* for the performance of humans supported by automated decision aids (Lee and Moray, 1992, 1994; Muir, 1987, 1994; Muir and Moray, 1996). The decision to either rely or not rely on automation can be one of the most important decisions a human operator can make, particularly in time-critical situations (Parasuraman and Riley, 1997). However, humans often fail to rely upon automation appropriately (Lee and See, 2004). Two potential problems are misuse and disuse (Parasuraman and Riley, 1997). Misuse refers to failures that occur when people inadvertently violate critical assumptions and rely on automation inappropriately, whereas disuse indicates failures that occur when people reject the capabilities of automation. Misuse and disuse are examples of inappropriate trust. Appropriate trust is when the trust someone has in another agent (human or computer) is in accordance with the capabilities of this agent.

Ideally humans rely on their own decisions when these are best and rely on the decision aid's when those are best. But operators do not base their reliance decisions on comparisons of true reliabilities of themselves and the decision aids. Rather, perceived reliabilities are usually imperfectly calibrated to true reliabilities, even after practice (van

Dongen and van Maanen, 2006). It is often found that humans rely either too much or too little on decision aids or themselves (Parasuraman and Riley, 1997; Skitka et al., 1999; Dzindolet et al., 1999; van Dongen and van Maanen, 2006). Recent work (van Maanen et al., 2007; van Dongen and van Maanen, 2006; van Maanen and van Dongen, 2005) also has shown it is possible for support systems to outperform humans in making appropriate reliance decisions.

Misuse and disuse can also occur in team context. A team is defined as two or more people with different tasks who cooperate to achieve specified and shared goals (Brannick et al., 1997). A team member often has to rely on various information sources, for example on another team member or incoming information from different systems. So, when working together in a team, inappropriate trust can endanger team performance. Team performance is concerned with the outcomes of the team on the task at hand.

In this paper we focus on human-computer teams with two people and two computers and all interaction is regulated through the computer interface. In this team context, two possible solution the already mentioned problems with inappropriate trust are explored. One solution tries to advice the human in making appropriate reliance decisions. It estimates the probable over- or under-trust someone has in different agents and then communicates this estimate. The other proposed solution also makes this estimate of over- and under-trust, but instead of letting the human decide what to do with it, the system takes over when it thinks the degree of over- and under-trust is above a certain criterion. This study investigated the effect these two solutions have on team performance. This investigation was done in a specific task environment related to classification of geographical areas by interpreting video footage from two Unmanned Aerial Vehicles (UAVs).

The paper is composed of the following sections. First, in Section 2 the generic support model is described on which the above two proposed solutions are based. The description of this generic model leads to several hypotheses which were tested by a series of experiments described in Section 3. The results are reported in Section 4. We conclude with a discussion in Section 5.

2 Reliance Support

2.1 Generic Support Model

We assume a *hybrid team* situation in which humans, decision aids and machines (s.a. airplanes) collaborate to achieve a certain task. An important factor influencing their collaboration is the degree of trust between the participants. *Trust* can be defined as the attitude towards another agent that the agent will help achieve its individual goals (Lee and See, 2004). Trust can be based on prior performance. In this study, for example, a UAV operator may not trust the automatic classification of the system because it made a mistake a moment ago. The trust a team member has in different agents guides his *reliance* on those agents, which can be defined as the act of trusting (Castelfranchi and Falcone, 1998). For example, if an *untrusted* classification system has classified an area as safe, the operator probably does not (but is able to) *rely* on this system and most probably will not automatically declare this area as a safety zone. If the operator would inappropriately rely on the classification system this would lead to errors. We call this situation *over-reliance*. If, on the other hand, the operator would choose not to rely on correct advice, an unnecessarily large amount of work would be imposed on the operator which could lead to errors as well. This situation is called *under-reliance*. In general, we can say that the more over- and under-reliance exists within a human-machine team, the more overall team performance diminishes. Preventing situations with over- and under-reliance is the purpose of the *reliance support system* described in this section. The generic architecture of this system is illustrated in Figure 1.

The system continuously monitors the human-machine team to collect data on who performs which actions under which circumstances with what success rate. This data forms the input of two processes which are simultaneously active: one process computing *actual reliance* and another process computing *optimal reliance*. Actual reliance can be estimated by taking into account the previous reliance behaviors of the participants. For example, if the operator has relied on its classification system in the past period of time, it is likely that he will continue to do so in the present situation. Optimal reliance can be computed by taking into account the past

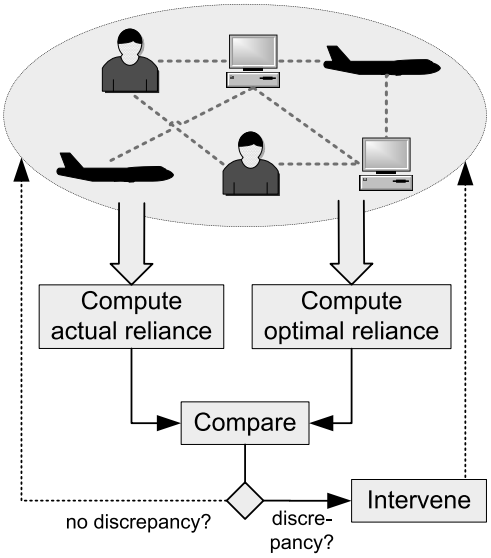


Figure 1. General model of reliance support system

performance of task performers. For example, if the classifications done by the automatic classification system in the recent past have been better than those done by the operator, we can infer that the optimal reliance behavior of the operator should be to rely on the classification system. If there is a discrepancy between the actual and the optimal reliance, the reliance support system will intervene. The purpose of the intervention is to repair occurrences of over- and under-reliance to improve team performance.

Of course, computing actual and optimal reliance is often more complicated, and can be done with different levels of sophistication and accuracy. Improving these models is a continuous effort, about which we have reported elsewhere (Hoogendoorn et al., 2010).

2.2 Proposed Support Types

As has been described in the introduction of this paper, we investigate two possible ways of intervention: one related to the communication of an estimate of over- and under-trust (from now on called graphical support (GS)) and the other related to taking over reliance decision making when this estimate is above a certain criterion (from now on called adaptive autonomy (AA)). These possible solutions are further explained in the following two paragraphs.

One possible solution tries to advice the human in making appropriate reliance decisions. It estimates the probable over- or under-trust someone has in different agents and then communicates this estimate. The other proposed possible solution also makes this estimate of over- and under-trust, but instead of letting the human decide what to do with it, the system takes over when it thinks the degree of over- and under-trust is above a certain criterion. This study investigated the effect these two solutions have on team performance. This investigation was done in a specific task environment related to classification of geographical areas by interpreting video footage from two Unmanned Aerial Vehicles (UAVs).

Graphical Support: Graphical support works by giving direct feedback about under- or over-reliance. For example, if the operator should rely more on his decision aid, an upward arrow is shown on his screen. If the operator should rely less on his decision aid, this can be visualized using a downward arrow. If there is no mis-calibration of reliance, the operator does not receive any graphical feedback. An instantiation of such graphical support is used in this study and is further described in Section 3.

Adaptive Autonomy: Another way to intervene in case of reliance mis-calibration, is to use adaptive autonomy (Dorais et al., 1998). Following this paradigm, the level of the system's autonomy is adjusted during system execution, depending on the current situation. For this purpose, we can distinguish three situations which we couple with corresponding levels of automation (Parasuraman et al., 2000). For instance, the difference between actual reliance behavior and optimal reliance behavior could be small, moderate or large. This difference can determine if the task should be allocated to the human or system. If the difference is small, the team member is able to carry out the task well, so the task is allocated to the human. If the difference is moderate, the human receives a certain time to veto the decision of the system. When the difference is large, the task is allocated to the system. An instantiation of such adaptive autonomy is used in this study and is further described in Section 3.

2.3 Hypotheses

Based on the above described generic model we propose several hypotheses about team performance, satisfaction and support effectiveness.

2.3.1 Team Performance: Due to the fact that there is a positive relation between appropriate trust and reliance for the performance of humans supported by decision aids (Lee and Moray, 1992, 1994; Muir, 1987, 1994; Muir and Moray, 1996), it is expected that both the graphical support and the adaptive autonomy will improve team performance. This results in the following hypothesis:

Hypothesis 1. *There is an increase of team performance for graphical support and adaptive autonomy compared to no support.*

2.3.2 Satisfaction: A potential problem in the proposed adaptive autonomy is satisfaction. As humans are less likely to accept others, and more specifically automation, to take over autonomy and therefore the responsibility for the appropriate outcome (i.e., locus of control; see Rotter, 1966), it is expected that the application of adaptive autonomy will result in less satisfaction. This leads to the following hypothesis:

Hypothesis 2. *Graphical support leads to a higher satisfaction than adaptive autonomy.*

2.3.3 Effectiveness due to Human Competence: It is expected that for humans with higher competence in task execution also more appropriate trust will occur. This can be explained by two reasons. The first reason is that higher competent humans also have an increased amount of cognitive resources left for calibrating trust appropriately. This would result in a lower need for assistance with respect to reliance decision making. This would on its turn lead to less attention going to a support system, trying to support reliance decision making. This will inevitably result in a decreased effect due to human competence, since the support system will simply be partly neglected. The second reason is that the way the proposed support system is designed results in less interventions by the system when it estimates that the human is trusting inappropriately. Since it has already been said that higher competence leads to more appropriate trust, higher competence will also lead to less interventions. And less interventions would also mean less effectiveness.

With respect to the difference in effectiveness between the graphical support and adaptive autonomy, we can say the following. Since adaptive autonomy occasionally takes over reliance decision making instead of only advising the human, there is not much to neglect for the human: the human is bypassed and his neglect has no effect on the effectiveness of adaptive autonomy. This results in the expectation that the decrease of effectiveness will be less for adaptive autonomy compared to the graphical support. However, also for adaptive autonomy the amount of interventions will be less, resulting in still a decrease of effectiveness due to human competence.

All of the above arguments lead to the statement that the inverse of human competence is actually a good predictor for the effectiveness of the different support types. This boils down to the following hypothesis:

Hypothesis 3. *Higher human competence leads to a decrease of effectiveness of graphical support and adaptive autonomy, though less decrease is expected for adaptive autonomy.*

2.3.4 Effectiveness due to Task Difficulty: Similar as for human competence, lower task difficulty also leads to an increase of available cognitive resources for calibrating trust and less interventions by the support system. This suggests the same type of influence of task difficulty on the effectiveness of the different support types. The hypothesis:

Hypothesis 4. *Low task difficulty leads to a decrease of effectiveness of graphical support and adaptive autonomy.*

Similar experiments from the literature also suggest the opposite of Hypotheses 3 and 4. In an experiment from, for instance, McGuirl and Sarter (2006), where dynamically changing confidence displays of system reliability were used, task performance was significant higher during low task load compared to high task load situations. But the difference between McGuirl and Sarter (2006)'s study and the present one is that the given support is only provided when the system estimates it is needed (i.e., mostly during periods of higher difficulty and

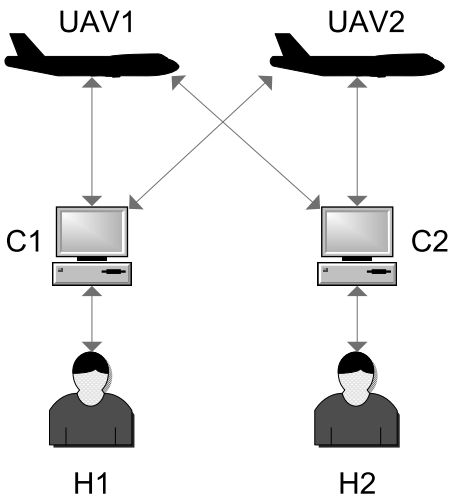


Figure 2. Experimental setup.

Table I
OBJECT TYPES AND THEIR SCORES.

Name	Image	Score
Tank		2
Rebel		1
Civilian		-1
Car		-2

Table II
DECISION CRITERIA TO CLASSIFY GEOGRAPHICAL AREAS AS AREAS THAT EITHER NEED TO BE ATTACKED, HELPED OR LEFT ALONE BY GROUND TROOPS.

≤ -3	-2	-1	0	1	2	$3 \leq$
Help area	Leave area alone					Attack area

low performance), which would result in an opposite effect.

3 METHOD

3.1 Participants

18 Participants (eight male and ten female) with an average age of 23 ($SD = 3.8$) participated in the experiment as paid volunteers. Participants were selected between the age of 20 and 30 and were not color blinded. All were experienced computer users, with an average of 16.2 hours of computer usage each week ($SD = 9.32$).

3.2 Apparatus

The experimental task was a classification task in which two participants on two separate personal computers (see Figure 2) had to classify geographical areas according to specific criteria as areas that either needed to be attacked, helped or left alone by ground troops. The participants needed to base their classification on real-time computer generated video images that resembled video footage of real UAVs. On the camera images, multiple objects were shown. There were four kinds of objects: civilians, rebels, tanks and cars. The identification of the number of each of these object types was needed to perform the classification. Each object type had a score (see Table I) and the total score within an area had be determined. Based on this total score and the decision criteria depicted in Table II, the participants could classify a geographical area (i.e., attack, help or do nothing). Participants had to classify two areas at the same time and in total 98 areas had to be classified. Both participants did the same areas with the same UAV video footage and were not allowed to talk to each other. The participants could indicate their choices via fixed keys on a computer keyboard.

During the time a UAV flew over an area, three phases occurred: The first phase was the *advice phase*. In this phase both participants and a decision aid gave an advice about the proper classification (attack, help or do nothing). This implies that there were three advices at the end of this phase. It was also possible for the participants to refrain from giving an advice, but this hardly ever happened. The second phase was the *reliance phase*. In this phase the advice of both the participants and the decision aid were communicated to each participant. Based on this advice the participants had to indicate which advice, and therefore which of the three trustees (self, other or decision aid), they trusted the most. Participants were instructed to maximize the number of correct classifications at both phases (i.e., advice and reliance phase). The third phase was the

feedback phase, in which the correct answer was given to both participants. Based on this feedback the participants could update their internal trust models for each trustee (self, other or decision aid).

In Figure 3 the interface of the task is shown. The map is divided in 10×10 areas. These boxes are the areas that were classified. The first UAV starts in the top left corner and the second one left in the middle. The UAVs fly a predefined route so participants do not have to pay attention to navigation. The camera footage of the upper UAV is positioned top right and the other one bottom right. The advice of the self, other and the decision aid was communicated via dedicated boxes below the camera images. The advice to attack, help or do nothing was communicated by red, green and yellow, respectively. On the overview screen on the left, feedback was communicated by the appearance of a green tick or a red cross. The reliance decision of the participant is also shown on the overview screen behind the feedback (feedback only shown in the feedback phase). The phase depicted in Figure 3 was the reliance phase before the participant indicated his reliance decision.

3.3 Design

A 3 (support type) $\times 2$ (task difficulty) within-subjects design was used. This means that every participant received every support type with two levels of difficulty. The order of support type was balanced between the participants in order to reduce effects of fatigue and practice. Three teams received the order NS–GS–AA, three teams the order GS–AA–NS and three teams AA–NS–GS (Latin square). For each support type, team performance and satisfaction was measured.

3.4 Independent Variables

There are two categorical independent variables: support type and task difficulty. Human competence is a continuous quasi-independent variable.

3.4.1 Support Type: Three levels of this independent variable are: 1) No support (NS), 2) Graphical Support (GS) and 3) Adaptive Autonomy (AA).

No Support (NS): For this support type no support is given with respect to the reliance decision the participant has to make. Support of the other participant and the decision aid in the form of advice is still given and does not alter between conditions (except when the task difficulty changes, both advices will have a higher probability to be less accurate).

Figure 4. Visual cues of the graphical support.

Graphical Support (GS): This support type assisted participants to correctly calibrate their trust in oneself, the other and the decision aid. The support indicated for each trustee S_1 whether the participant is expected to over- or under-trust S_1. The graphical support changed dynamically based on recent information about the reliance behavior of the participant and the performance of the three trustees. As monitoring dynamic information can be a cognitively demanding (Bartram et al., 2003), the support is based on simple visual cues (see Figure 4). The direction of the arrow indicates whether a person is advised to rely less or more on either of the trustees. If no arrow is visible, no change of reliance behavior is adviced.

After each feedback phase the graphical cues based on the estimation of the appropriateness of trust are updated. Trust is defined as appropriate when instances of over- and under-trust is within certain limits. Trust appropriateness is calculated in the following manner: First it is estimated what the current trust of the participant in the different trustees is (using Hoogendoorn et al., 2008). This type of trust is called 'descriptive trust', indicated by $\tau_i^{\mathrm{d}}(t)$ for trustee S_i at time point t, which has a value between 0 (no trust) and 1 (maximum trust). Second it is estimated what the trust would be of a rational agent in the different trustees (using van Maanen et al., 2007; Jonker and Treur, 1998). This type of trust is called 'prescriptive trust', indicated by $\tau_i^{\mathrm{p}}(t)$ for trustee S_i at time point t, which also has a value between 0 and 1.[1] Trust appropriateness is then calculated by the equation:

[1] A detailed explanation of descriptive and prescriptive models of trust is not in the scope of this paper. Those further interested are referred to the mentioned papers.

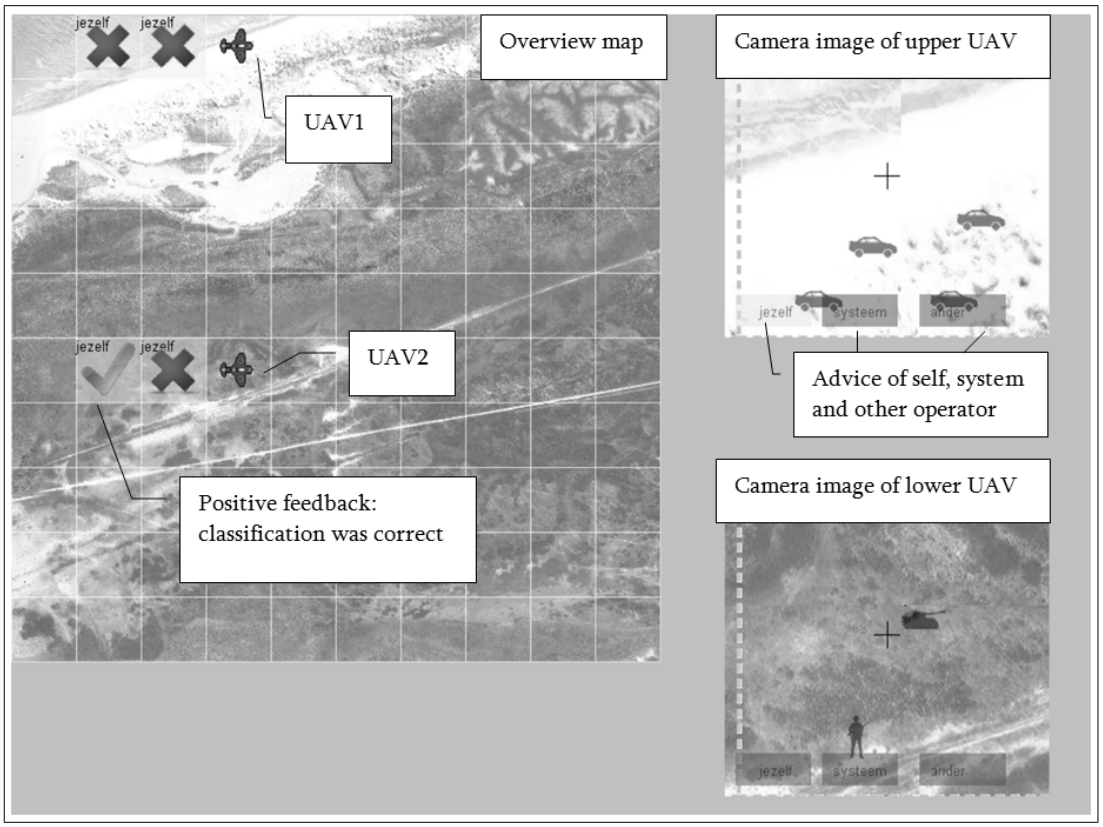


Figure 3. Interface of the task.

$$\alpha_i(t) = \tau_i^{\mathrm{d}}(t) - \tau_i^{\mathrm{p}}(t)$$

with $-1 \leq \alpha_i(t) \leq 1$ for trustee S_i at time point t. Positive trust appropriateness values indicate over-trust and negative values under-trust. When it holds that $|\alpha_i(t)| \leq .08$ then no arrow is displayed, when $\alpha_i(t) > .08$ an upward arrow is displayed and a downward arrow otherwise (i.e., when $\alpha_i(t) < -.08$). In order to be certain that interventions occurred, the .08 threshold was chosen by calculating the average absolute value of trust appropriateness during a pilot, which is equal to $\sum_{t=1}^{t_e} \frac{|\alpha_i(t)|}{t_e}$, where $t_e = 49$ (the number of feedback phases during an experiment).

Adaptive Autonomy (AA): This support type made use of three levels of autonomy (LOAs) which are applied dynamically during the task. The used LOAs are shown in Table III.

Table III
LEVELS OF AUTONOMY (LOAS) BASED ON ESTIMATED APPROPRIATENESS OF TRUST.

Trust appropriateness	Level of autonomy (LOA)
Appropriate	LOA1: manual
Less appropriate	LOA2: management-by-execution
Not appropriate	LOA3: autonomous

The different LOAs were triggered in a similar way as the graphical support: When it held that $\sum_i |\alpha_i(t)| \leq 0.2$, then the reliance decision was made by the participant during the reliance phase (LOA1: manual). When it held that $.2 \leq \sum_i |\alpha_i(t)| \leq 0.25$, then the participant

Figure 5. Visual cues of the adaptive autonomy.

was able to indicate his or her reliance decision, but was required to confirm his decision by pressing a confirmation key, otherwise the support determined the reliance decision (LOA2: management-by-execution). When it held that $.25 \leq \sum_i |\alpha_i(t)|$, then the support always made the reliance decision (LOA3: autonomous). In both LOAs 3 and 2, when the user did not react before the end of the reliance phase, the decision of the support was used as reliance decision. The current LOA of the support was indicated by a visual cue on the interface of the task (see Figure 5), where a square around a large image of a computer indicated that LOA3 was selected and a square around a small image of a computer indicated that LOA1 was selected.

The reliance decision of the support was based on the advice of trustee S_i for which it held that $\tau_j^{\mathrm{p}}(t) \leq \tau_i^{\mathrm{p}}(t)$ for all trustees S_j.[2]

3.4.2 Human Competence: The second independent variable was human competence. This variable is quasi-independent because human competence was determined by the task performance of the participant in the NS condition. This task performance was calculated by averaging the penalties given for the final decisions in each reliance phase during the NS experiment. See Equation 1 in Section 3.5 for this calculation and its explanation. Human competence was used as a predictor for the difference in team performance when applying the different support types. In pilots, no significant learning effects were found for human competence, which allowed us to use the NS condition in spite of the fact that the order of the support types was balanced between subjects.

3.4.3 Task Difficulty: In order to test the effect of task difficulty on the increased effect of support type on team performance, task difficulty was altered halfway each support type condition (after 50 classifications). Task difficulty had two levels. The first part of the experiment was easy and the second part difficult. In the difficult part, objects (cars, rebels, civilians and tanks) were partially camouflaged so that they blended into the surroundings. This was done by changing the alpha-value (transparency) of the images. Also, the number of objects in an area and the number of different objects was increased for the difficult part. The easy part contained on average 3 objects and 1.66 different objects, whereas the difficult part contained on average 6.5 objects and 2.5 different objects.

Furthermore, the reliability of the decision aid was controlled within the easy and difficult part. Decision aid reliability was a control variable within the easy and difficult parts and was not used as an independent variable. On average the reliability of the decision aid was 80% (first part 75% en then 85%) in the easy and 70% (first part 65% and then 75%) in the difficult. This was done in order to decrease the effect of non-triviality in determining which trustee would be best to trust: i.e., task performances between trustees would be more equal and probability to rely on either one of the different trustees would be more equalized.

3.5 Dependent Variables

The dependent variables were team performance and satisfaction.

3.5.1 Team Performance: Team performance was based on the average penalties given over the final decisions in each reliance phase. There were several situations in which either the participant him- or herself made the final decision, or it was the adaptive autonomous support that made the final decision. In the NS and GS conditions, it was always the participant who made the final decision. In the AA condition, only when LOA1 or LOA2 was selected the participant made the final decision, except for LOA2 when the reliance decision was not confirmed by the participant (i.e., by pressing the confirmation key). In the case that this decision was indeed not confirmed, in LOA2 the support system took over and made the final decision. When LOA3 was selected, the final decision was always made by the support system. Because of this mixed initiative situation and because the final decision was also based on the advice of the different team members

[2]Selection of LOAs can be perceived as a meta-reliance decision, i.e., the support system decides to rely either on the support or the participant with respect to their reliance decisions.

(human or machine), the measured performance is called *team* performance, i.e., the final decision is not only made by the participant him- or herself or based on his or her own opinion.

As mentioned, the team performance was calculated based on an average of penalties. The penalty $p_i(x)$ for each area x was calculated as follows: Let $d_i(x) = 0$ when the final decision for area x was 'help', $d_i(x) = .5$ when it was 'do nothing' and $d_i(x) = 1$ when it was 'attack'. Similarly, let $a_i(x) = 0$, .5 or 1 when the answer given in the feedback phase was 'help', 'do nothing' or 'attack', respectively. Then it held that $p_i(x) = |d_i(x) - a_i(x)|$, with $p_i(x) = 1$ being the worst and $p_i(x) = 0$ being the best final decision for area x. The idea behind this was that attacking while it was necessary to help was worse than attacking while one did not need to do anything. Similarly, it was worse to help when actually an attack was needed than to help while nothing needed to be done. Finally, to decide to do nothing was a fairly safe decision since this always resulted in $p_i(x) \leq 0.5$. When no decision was made a penalty of 1 is awarded; so to decide to do nothing was different from actually doing nothing with respect to the final decision.

Based on the above, team performance was calculated by the following equation:

$$P_i = \frac{x_e - \sum_{x=1}^{x_e} p_i(x)}{x_e} = 1 - \sum_{x=1}^{x_e} \frac{|d_i(x) - a_i(x)|}{x_e} \quad (1)$$

where x_e was the number of the last area in the experiment, which was equal to 98.

3.5.2 Satisfaction: Participants rated after the GS and AA condition the degree to which they thought the support system was satisfactory on a 5-point Likert scale between 1 (terrible) and 5 (fantastic).

3.6 Procedure

Participants were given thorough instructions about the details given in Section 3.2. The understanding of participants' knowledge about the classification was tested by means of eight assignments. A minimum of six out of eight had to be correct or otherwise a re-examination with eight different examples was done.

Furthermore, for each condition the participants were able to get used to the interface and support types. After this, the participants' know-how was tested by means of seven multiple-choice questions about the task. This test was reviewed by the experimenter together with the participants.

In total the experiment took 110 minutes. Each support type condition took 10 minutes. There was a break of 5 minutes after each support type condition. Giving instructions, filling in forms and doing exercises cost 55 minutes in total. An additional NS condition (10 minutes) was done for each participant for the purpose of personalizing and optimizing the parameters of the trust models used by the support types (see Hoogendoorn et al., 2010) during the break.

4 Results

4.1 Team Performance

In Figure 6 the main effect of support type (either no support (NS), graphical support (GS) or adaptive autonomy (AA)) for team performance is shown. A repeated measures analysis of variance (ANOVA) showed no significant main effect ($F(2, 24) = 2.0176$, $p = .15$). This means that based on the data from this experiment no evidence is found for increase of team performance for GS ($M = 0.8941$, $SD = 0.0450$) and AA ($M = 0.8695$, $SD = 0.0534$) compared to NS ($M = 0.8721$, $SD = 0.0679$). Hence Hypothesis 1 is not accepted.

4.2 Satisfaction

A Wilcoxon Signed-ranks test indicated that GS was more satisfactory ($Mdn = 3$) than AA ($Mdn = 2$), $Z = 2.24$, $p = .02$. Hence Hypothesis 2 is accepted.

4.3 Effectiveness due to Human Competence

Figure 7 shows the regression lines after linear regression on the increase of team performance of GS compared to NS (top), AA compared to NS (middle) and AA compared to GS (bottom), with human competence as predictor. Human competence was a highly significant predictor for the increase of team performance of GS compared to

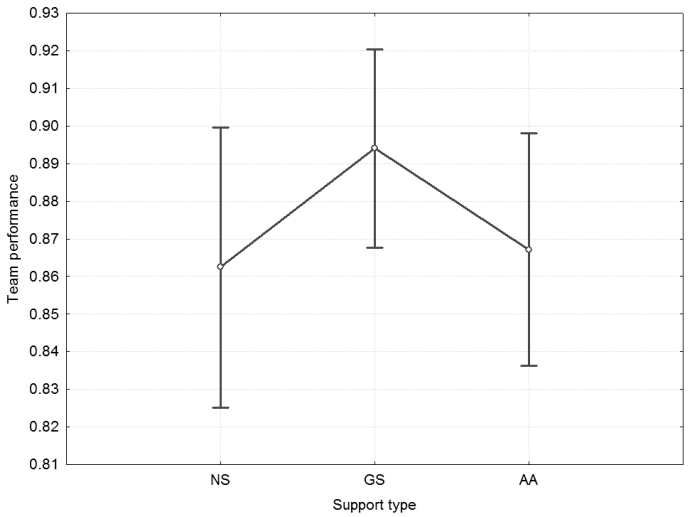


Figure 6. Main effect of support type for team performance.

NS ($\beta = -.76$, $p = .002$), AA compared to NS ($\beta = -.74$, $p = .003$), but not for AA compared to GS ($\beta = -.13$, $p = .65$). In other words, this shows that higher human competence indeed leads to a decrease of effectiveness of the different support types, and therefore Hypothesis 3 is accepted, except for AA compared to GS.

4.4 Effectiveness due to Task Difficulty

Figure 8 shows the possible interaction effect between task difficulty (low or high difficulty) and support type comparisons on the increase of team performance. In other words, this figure shows whether higher task difficulty leads to larger differences of team performance for GS compared to NS, AA compared to NS and AA compared to GS. No significant interaction effect was found ($F(2, 52) = 0.67$, $p = .52$). Hence higher task difficulty does not lead to a higher increase of team performance for both GS and AA as compared to NS and therefore Hypothesis 4 is not accepted.

5 Discussion and Conclusions

Given that many studies have shown convincing evidence for the importance of trust in performance of humans supported by decision aids (Lee and Moray, 1992, 1994; Muir, 1987, 1994; Muir and Moray, 1996) and that humans often fail to rely upon automation appropriately (Lee and See, 2004; Parasuraman and Riley, 1997), the development of intelligent systems supporting human reliance decision making seems promising. Main research goal of this study was to find out if two types of such support would indeed result in an increase of human-decision aid team performance. Team performance in the support conditions were somewhat higher compared to no support. However, these differences were not significant.

The results of using graphical support can be compared to, for instance, the results of McGuirl and Sarter (2006) (though the task and support type are different) where confidence information about system reliability increased both task performance and self-reported accuracy of the estimation of current system reliability. For correct performance users needed visual as well as kinesthetic cues. For this task only visual cues were available and a possible limitation of the graphical support could be explained by single modularity interferences. It may have been more difficult for the participants to pay attention to visual task information as well as support information at the same time. Possible future variants of reliance decision support should

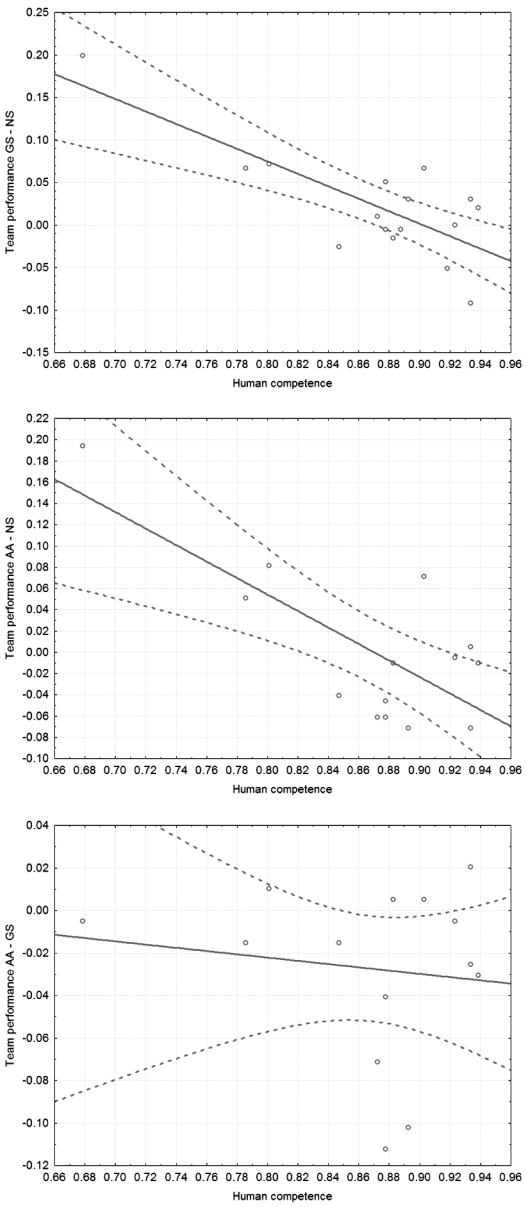


Figure 7. Regression lines for the increase of team performance of GS compared to NS (top), AA compared to NS (middle) and AA compared to GS (bottom), with human competence as predictor.

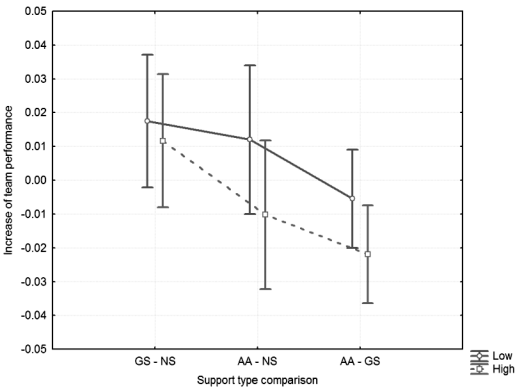


Figure 8. Increase of team performance for GS compared to NS, for AA compared to NS and for AA compared to GS, for low and high task difficulty.

therefore aim at making the interpretation of the support less intrusive.

As mentioned, the results of using adaptive autonomy also did not show a significant improvement compared to no support. Parasuraman et al. (1996)'s results have shown that performance-based allocation of tasks can improve monitoring of automation. The difference between Parasuraman et al. (1996)'s and the present study is that the trigger for support is the estimated performance of trust calibration instead of task performance. This trust calibration performance estimate may not have been accurate enough for enough effective interventions. We feel strengthened by the fact that indeed taking over reliance decisions by the computer can lead to significant performance improvement (van Maanen et al., 2007) and therefore future research should also focus on the validity of trust models used by the support. Improving these models is a continuous effort, about which we have reported elsewhere (Hoogendoorn et al., 2010). Furthermore, results showed that satisfaction with adaptive autonomy compared to graphical support was lower, which could suggest that there was also a decrease of performance due to a decrease of dedication to the task. Future research should also aim at investigating new efforts for taking away reasons for, for instance, human intolerance for increased machine autonomy in making (important) decisions.

Another reason for the found insignificant effect

of the investigated support types could be the fact that also no significant effect was found between the reliance performance of the operator and the system.[3] It might be the case that the task to make reliance decisions was too easy. This in spite of the effort to design the experiment in such a way that it was not trivial for the participants to determine which trustee would be best to trust. This was to overcome floor and ceiling effects when it comes to reliance on the different trustees (e.g., one would almost always count on one's computer for, for instance, the multiplication of two large numbers; this reliance decision is simply to easy to be supported for). The significant results of the decrease of effectiveness due to human competence could also suggest that once task performance becomes too high, the reliance decision becomes more easy and therefore the potential effectiveness of the support decreases. Future efforts should aim at investigating what precisely goes wrong when making reliance decisions, why this is such a difficult task for humans and how to provide leverage for exactly that.

Finally, the triggering of adaptive support was based on trust estimation and in spite of the fact that trust is such an important factor influencing team performance, there are also other factors that mediate the relationship between human beliefs and their reliance behavior (Lee and See, 2004): e.g., psychological and environmental factors that have not been used here. Further research should investigate whether it is of benefit for adaptive team support to include such factors.

Acknowledgments

This research was partly funded by the Dutch Ministry of Defense under program number V929.

References

Bartram, L., Ware, C., and Calvert, T. (2003). Moticons: detection distraction and task. *International Journal of Human-Computer Studies*, 58(5):515–545.

Brannick, M. T., Salas, E., and Prince, C., editors (1997). *Team Performance Assessment and Measurement: Theory, Methods, and Applications*. Lawrence Erlbaum Associates, Mahwah, NJ.

Castelfranchi, C. and Falcone, R. (1998). Principles of trust for MAS: Cognitive anatomy, social importance, and quantification. In *Proceedings of 3rd International Conference on MultiAgent Systems*, pages 72–79.

Dorais, G. A., Bonasso, R. P., Kortenkamp, D., Pell, B., and Schreckenghost, D. (1998). Adjustable autonomy for human-centered autonomous systems on mars. In *Proceedings of the First International Conference of the Mars Society*.

Dzindolet, M. T., Pierce, L. G., Beck, H. P., and Dawe, L. A. (1999). Misuse and disuse of automated aids. In *Proceedings of the Human Factors Society 43rd Annual Meeting*, pages 339–343, Santa Monica, CA.

Grootjen, M. and Neerincx, M. (2005). Operator load management during task execution in process control. In *Human Factors Impact on Ship Design*.

Hoogendoorn, M., Jaffry, S., and Treur, J. (2008). Modeling dynamics of relative trust of competitive information agents. In Klusch, M., Pechoucek, M., and Polleres, A., editors, *Proceedings of the 12th International Workshop on Cooperative Information Agents (CIA'08)*, volume 5180 of *LNAI*, pages 55–70. Springer.

Hoogendoorn, M., Jaffry, S. W., and van maanen, P.-P. (2010). Validation of agent models of trust: Independent compared to relative trust. Submitted to conference.

Jonker, C. M. and Treur, J. (1998). Formal analysis of models for the dynamics of trust based on experiences. In Garijo, F. J. and Boman, M., editors, *Multi-Agent System Engineering, Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW'99*, volume 1647, pages 221–232, Berlin. Springer Verlag.

Lee, J. and Moray, N. (1992). Trust, control strategies, and allocation of function in human-machine systems. *Ergonomics*, 35:1243–1270.

Lee, J. and Moray, N. (1994). Trust, self-confidence, and operators' adaption to automation. *International Journal of Human-Computer Studies*, 40:153–184.

Lee, J. D. and See, K. A. (2004). Trust in automa-

[3] These specific results have been left out of this paper for reasons of brevity.

tion: Designing for appropriate reliance. *Human Factors*, 46(1):50–80.

McGuirl, J. M. and Sarter, N. B. (2006). Supporting trust calibration and the effective use of decision aids by presenting dynamic system confidence information. *Human Factors*, 48(4):656–665.

Muir, B. M. (1987). Trust between human and machines, and the design of decision aids. *International Journal of Man-Machine Studies*, 27(5-6):527–539.

Muir, B. M. (1994). Trust in automation: Part i. theoretical issues in the study of trust and human intervention in automated systems. *Ergonomics*, 37(11):1905–1922.

Muir, B. M. and Moray, N. (1996). Trust in automation, part ii. experimental studies of trust and human intervention in a process control simulation. *Ergonomics*, 39(3):429–460.

Parasuraman, R., Mouloua, M., and Molloy, R. (1996). Effects of adaptive task allocation on monitoring of automated systems. *Human Factors*, 38(4):665–679.

Parasuraman, R. and Riley, V. A. (1997). Humans and automation: Use, misuse, disuse, abuse. *Human Factors*, 39:230–253.

Parasuraman, R., Sheridan, T. B., and Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics*, 30:286–297.

Rotter, J. B. (1966). General expectancies for internal versus external control of reinforcement. *Psychological Monographs: General and Applied*, 80(1). Whole no. 609.

Skitka, L. J., Mosier, K. L., and Burdick, M. (1999). Does automation bias decision-making? *International Journal of Human-Computer Studies*, 51(5):991–1006.

van Dongen, K. and van Maanen, P.-P. (2006). Under-reliance on the decision aid: A difference in calibration and attribution between self and aid. In *Proceedings of the Human Factors and Ergonomics Society's 50th Annual Meeting*.

van Maanen, P.-P., Klos, T., and van Dongen, K. (2007). Aiding human reliance decision making using computational models of trust. In *Proceedings of the Workshop on Communication between Human and Artificial Agents (CHAA'07)*, pages 372–376, Fremont, California, USA. IEEE Computer Society Press. Co-located with The 2007 IEEE IAT/WIC/ACM International Conference on Intelligent Agent Technology.

van Maanen, P.-P. and van Dongen, K. (2005). Towards task allocation decision support by means of cognitive modeling of trust. In Castelfranchi, C., Barber, S., Sabater, J., and Singh, M., editors, *Proceedings of the Eighth International Workshop on Trust in Agent Societies (Trust 2005)*, pages 168–77.

Part III

Attention

Chapter 9

Augmented Meta-Cognition Addressing Dynamic Allocation of Tasks Requiring Visual Attention

This chapter appeared as (Bosse et al., 2007b).

Augmented Meta-Cognition addressing Dynamic Allocation of Tasks Requiring Visual Attention

Tibor Bosse*, Willem van Doesburg†, Peter-Paul van Maanen*† and Jan Treur*

* Department of Artificial Intelligence, Vrije Universiteit Amsterdam
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
Email: {tbosse, treur}@cs.vu.nl
† TNO Human Factors, P.O. Box 23, 3769 ZG Soesterberg, The Netherlands
Email: {willem.vandoesburg, peter-paul.vanmaanen}@tno.nl

***Abstract*—This paper discusses the use of cognitive models as augmented meta-cognition on task allocation for tasks requiring visual attention. In the domain of naval warfare, the complex and dynamic nature of the environment makes that one has to deal with a large number of tasks in parallel. Therefore, humans are often supported by software agents that take over part of these tasks. However, a problem is how to determine an appropriate allocation of tasks. Due to the rapidly changing environment, such a work division cannot be fixed beforehand: dynamic task allocation at runtime is needed. Unfortunately, in alarming situations the human does not have the time for this coordination. Therefore system-triggered dynamic task allocation is desirable. The paper discusses the possibilities of such a system for tasks requiring visual attention.**

***Index Terms*—Visual Attention, Cognitive Modeling, Augmented Cognition.**

1 Introduction

The term *augmented cognition* (Horvitz et al., 2001; Schmorrow and Kruse, 2004) was used by Eric Horvitz at the ISAT Woods Hole meeting in the summer of 2000 to define a potentially fruitful endeavor of research that would explore opportunities for developing principles and computational systems that support and extend human cognition by taking into explicit consideration well-characterized limitations in human cognition, spanning attention, memory, problem solving, and decision making. This paper focuses on extending human cognition by the development of principles and computational systems addressing task allocation of tasks requiring visual attention. In previous work (Bosse et al., 2006), cognitive models of visual attention were part of the design of a software agent that supports a naval warfare officer in its task to compile a tactical picture of the situation in the field. In the domain of naval warfare, the complex and dynamic nature of the environment makes that the warfare officer has to deal with a large number of tasks in parallel. Therefore, in practice, (s)he is often supported by software agents that take over part of these tasks. However, a problem is how to determine an appropriate allocation of tasks: due to the rapidly changing environment, such a work division cannot be fixed beforehand (Bainbridge, 1983). Task allocation has to take place at runtime, dynamically. For this purpose, two approaches exist, i.e., *human-triggered* and *system-triggered* dynamic task allocation (Campbell et al., 1997). In the former case, the user can decide up to what level the software agent should assist her. But especially in alarming situations the user does not have the time to think about such task allocation (Inagaki, 2003). In these situations it would be better if a software agent augments the user's meta-cognitive capabilities by means of system-triggered dynamic task allocation. This paper discusses the usage of cognitive models of visual attention that can be incorporated within assisting software agents offering augmented meta-cognition in order to obtain such a system-triggered dynamic task allocation,

In Section 2 a further elaboration on the motivational background for augmented meta-cognition is given. Section 3 a generic design of augmented meta-cognition based on cognitive models of visual attention is described. In Section 4 some applica-

tions of the framework are introduced and discussed. The paper is concluded with a general discussion and some future research.

2 Augmented Meta-Cognition: Motivational Background

Support of humans in critical tasks may involve a number of aspects. First, a software agent can have knowledge about the task or some of its subtasks and, based on this knowledge, contribute to task execution. Usually, performing this will also require that the software agent has knowledge about the environment. This situation can be interpreted as a specific form of augmented cognition: *task-content-focused augmented cognition*. This means that the cognitive capabilities to do the task partly reside within the software agent, external to the human, and may extend the human's cognitive capabilities and limitations. For example, if incoming signals require a very fast but relatively simple response, in speed beyond the cognitive capabilities of a human, a software agent can contribute to this task, thus augmenting the human's limited reaction capabilities. Another example is handling many incoming stimuli at the same time, which also may easily be beyond human capabilities, whereas a software agent can take care of it.

If the software agent provides task-content-focused augmented cognition, like in the above two examples, it may not have any knowledge about the coordination of the subtasks and the process of cooperation with the human. For example, task allocation may completely reside at the human's side. However, as discussed in the introduction, when the human is occupied with a highly demanding task, the aspect of coordination may easily slip away. For example, while working under time pressure, humans tend to spend less attention to reflection on their functioning. If the software agent detects and adapts to those situations it will have a beneficial effect (e.g., Kaber and Endsley, 2004). This type of reflection is a form of meta-cognition: cognitive processes addressing other cognitive processes. A specific type of support of a human from an augmented cognition perspective can also address such reflective aspects: *augmented meta-cognition*. This is the form of augmented cognition that, in contrast to task-content-focused augmented cognition, addresses the support or augmentation of a human's limitations in meta-cognitive capabilities. The type of augmented meta-cognition discussed in this paper focuses on dynamic task allocation.

Augmented meta-cognition can be provided by the same software agent that provides task-content-focused augmented cognition, or by a second software agent that specializes on meta-cognition, for example the task allocation task. The former case results in a reflective software agent that has two levels of internal processing: it can reason both about the task content (object-level process) and about the task coordination (meta-level process) (e.g., Maes and Nardi, 1988). The latter case amounts to a specific case of a reflective multi-agent system: a multi-agent system in which some of the agents process at the object level and others at the meta-level.

The distinction made between task-content-focused augmented cognition and augmented meta-cognition provides a designer with indications for structuring a design in a transparent manner, either by the multi-agent system design, or by the design of a reflective software agent's internal structure. This paper focuses on the latter, the design of the reflective internal structure of the software agent. An implementation of such an agent has been evaluated for two case studies.

3 Augmented Meta-Cognition Design

In this section, first the generic design of the proposed augmented meta-cognition is presented in Section 3.1. After that, Section 3.2 describes how principles of Signal Detection Theory (SDT) can be applied within this design.

3.1 Prescriptive and Descriptive Models

The present design is based on the idea that the software agent's internal structure augments the user's meta-cognitive capabilities. This structure is composed of two maintained models of the user's attention. The first is called a *descriptive model*, which os a model that estimates the user's actual attentional dynamics. The second is called a *prescriptive model*, which prescribes the way these dynamics should be. In Figure 1 a conceptual design of such a software agent is shown. Depending on the user's and the agent's own attentional levels,

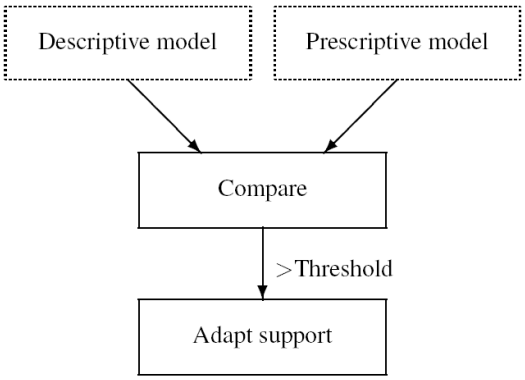


Figure 1. Conceptual model of the attention allocation system.

the agent decides whether the user (or the agent itself) is paying enough attention to the right tasks at the right time. This is determined by checking whether the difference between described attention and prescribed attention is below a certain threshold. In Figure 1 this comparison is depicted in the middle as the *compare* process. Based on this, the agent either adapts its support or it does not, i.e., the *adapt* process in Figure 1.

From the perspective of the agent, the runtime decision whether to allocate a task to itself or to the user comes down to the decision whether to support this task or not. The question remains what the agent could use as a basis for deciding to take over responsibility of a task, i.e., by exceedance of a certain threshold, using both descriptive and prescriptive models of user attention. An answer to this is that the agent's decision to support can be based on several performance indications: ($PI1$) a performance indication of the user concerning her ability to appropriately allocate attention (to the right tasks at the right time), ($PI2$) a performance indication of the agent concerning its ability to soundly prescribe the allocation of attention to tasks, ($PI3$) a performance indication of the system concerning its ability to soundly describe the user dynamics of the allocation of attention to tasks, and ($PI4$) a performance indication of the agent concerning its ability to soundly decide to support the user in her task to allocate attention to tasks.

3.2 Some Principles of SDT

One of the ways to let the agent estimate the performances of the user and the agent itself from the previous paragraph is by using the principles of *Signal Detection Theory*, or simply SDT (Green and Swets, 1966). In this subsection a theoretical framework based on SDT is defined, including a method that constitutes a means for identifying when to trigger attention allocation support.

To let a software agent reason about the performance of the user concerning her ability to appropriately allocate attention ($PI1$), a formal framework in SDT terms is needed in which it can describe it. These terms are mainly based on a mathematical description of the following situations:

1) The descriptive model of user attention indicates that attention is paid (A) to the tasks that are required by the prescriptive model (R). This situation is also called a *hit* (HIT).
2) The descriptive model of user attention indicates that attention is not paid (A^C, i.e., not A) to the tasks that are not required by the prescriptive model (R^C). This situation is also called a *correct rejection* (CR).
3) The descriptive model of user attention indicates that attention is paid (A) to the tasks that are not required by the prescriptive model (R^C). This situation is also called a *false alarm* (FA).
4) The descriptive model of user attention indicates that attention is not paid (A^C) to the tasks that are required by the prescriptive model (R). This situation is also called a *miss* ($MISS$).

The task to discriminate the above situations can be set out in a table as a 2-class classification task. The specific rates of HITs, FAs, $MISS$s, and CRs, are calculated by means of probabilities of the form $\Pr(X \mid Y)$, where X is the estimate of certain behavior and Y is the estimate of the type of situation at hand. The descriptive and prescriptive models mentioned earlier can be seen as the user's attentional behavior (A or A^C) in a specific situation that either requires attention (R) or does not (R^C). A HIT, for example, would be in this case $\Pr(A \mid R)$, and a FA would be $\Pr(A \mid R^C)$, etc. This classification task is shown in Table I. A similar task can

be defined for the other performance indicators, i.e., $PI2$, $PI3$, and $PI4$.

In SDT, the measure of sensitivity (d') is commonly used as an indicator for various kinds of performances. The measure is a means to compare two models, in this case descriptive and prescriptive models. Hence the calculation of such sensitivities can be used by the agent to determine whether to support the user or not. For instance. low sensitivities ($<$ threshold) may result in the decision to adapt support. The calculation of sensitivity in terms of the above mentioned HIT, FA, $MISS$, and CR, can be done by using the following formula:

$$d' = HIT - FA = CR - MISS \qquad (1)$$

As can be seen in equation 1, to calculate sensitivity, the measurement of HIT and FA are sufficient. No estimates of CR or $MISS$ are needed, since $HIT - FA$ is equal to $CR - MISS$.[1] Furthermore, sensitivity is dependent on both HIT and FA, rather than on HIT or FA alone. A user that has a high sensitivity as a result of attending to all tasks all the time (high HIT rate), is not only impossible due to the maximum capacity of human attention, but also very inefficient. Think of the very limited attention each task probably will receive due to unneeded FAs. The other way around, a low FA rate as a result of attending to nothing, is obviously not desired as well.

4 Applications

This section discusses two applications of the presented framework for task allocation based on visual attention. In Section 4.1, a pilot study is described, of which the main aim was to establish a (descriptive) model of a person's visual attention in the execution of a simple task in the warfare domain. For this pilot study, a simplified version of an Air Traffic Control (ATC) task was used. Next, Section 4.2 addresses a more realistic case: the task of Tactical Picture Compilation (TPC) by a naval warfare officer. For both cases, it is explained how descriptive models of visual attention may be used for task allocation, using the design introduced in Section 3.

[1] This is due to the fact that $HIT = 1 - MISS$ and therefore a high HIT results in a low $MISS$, and vice versa (the same holds for $CR = 1 - FA$).

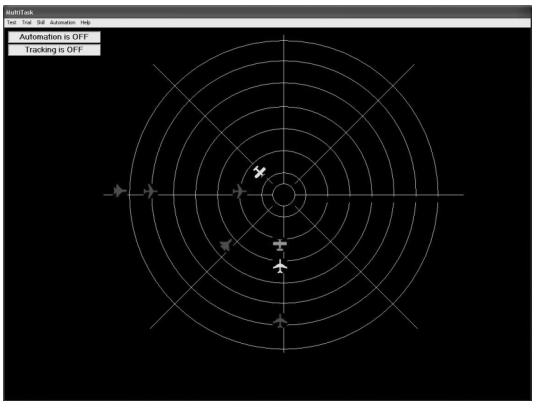


Figure 2. The interface of the used environment based on Multi-Task (Clamann et al., 2002).

4.1 Multitask

In order to test the ideas presented in the previous sections, a pilot study has been performed. The setup of this pilot study consisted of a human participant executing a simple warfare officer-like task (Bosse et al., 2006). To create such a setup, the software Multitask (Clamann et al., 2002) was used (and slightly altered in order to have it output the proper data). Multitask was originally meant to be a low fidelity ATC simulation. In this study, it is considered to be an abstraction of the cognitive tasks concerning the compilation of the tactical picture, i.e., a warfare officer-like task. A screen-shot of the task is shown in Figure 2.

In the pilot case study, the participant (controller) had to manage an airspace by identifying aircrafts that all are approaching the center of a radarscope. The center contained a high value unit (HVU) that had to be protected. In order to do this, airplanes needed to be cleared and identified to be either hostile or friendly to the HVU. The participant had to click on the aircraft according to a particular procedure depending on the status of the aircraft. Within the conducted pilot study, three different aircraft types were used, which resulted in different intervals of speed of the aircrafts. The above dynamic properties of the environment were stimuli that resulted in constant change of the participant's attention. The data that were collected consist of all locations, distances from the center, speeds, types,

Table I
A 2-CLASS CLASSIFICATION TASK BASED ON A DESCRIPTIVE (ATTENTION ALLOCATED) AND PRESCRIPTIVE (ATTENTION REQUIRED) MODEL OF USER ATTENTION.

		Attention required?	
		Yes	*No*
Attention allocated?	*Yes*	$HIT = \Pr(A \mid R)$	$FA = \Pr(A \mid R^C)$
	No	$MISS = \Pr(A^C \mid R)$	$CR = \Pr(A^C \mid R^C)$

and states (i.e., colors). Additionally, data from a Tobii x50 eye-tracker[2] were extracted while the participant was executing the task. All data were retrieved several times per second (10-50 Hz).

Based on such data, a cognitive model has been implemented that estimates the distribution of the user's attention over the locations of the screen at any moment during the experiment (Bosse et al., 2006). This model uses two types of input, i.e., *user-input* and *context-input*. The user-input is provided by the eye-tracker, and consists of the (x, y)-coordinates of the gaze of the user over time. The context-input is provided by the Multitask environment, and consists of the variables speed, distance to the center, type of aircraft, and aircraft status. The output of the model is represented in the form of an dynamically changing 3D image. An example screen-shot of this is shown in Figure 3 at an arbitrary time point.[3] The x-and y-axis denote the x- and y-coordinates of the grid, and the z-axis denotes the level of attention. In addition, the locations of all tracks, the status of the tracks, the location of the gaze, and the mouse clicks are indicated in the figure by small dots, color, a star, and a big dot, respectively. Figure 3 clearly shows that at this time point there are two peaks of attention (locations $(12, 10)$ and $(16, 9)$). Moreover, a mouse click is performed at location $(16, 9)$, and the gaze of the subject is also directed towards that location.

In terms of Section 3, the presented model is a descriptive model of the task. If, in addition to this a prescriptive model is created, both models can be used for dynamic task allocation, using the principles of Signal Detection Theory. Hence the presented model of visual attention can be used for augmented meta-cognition purposes: the system maintains a cognitive model of the attentional dynamics of an user, and accordingly, extends the user's meta-cognitive capabilites. By introducing a threshold, a binary decision mechanism can be established, which decides for each location whether it receives (enough) attention or not (A or A^C in Table I). The idea is to use such a mechanism for dynamic task allocation for the type of tasks in the naval domain as considered in this paper. For example, in case an user is already allocated to some task, it may be better to leave that task for him or her, and allocate tasks to the system for which there is less or no commitment from the user (yet).

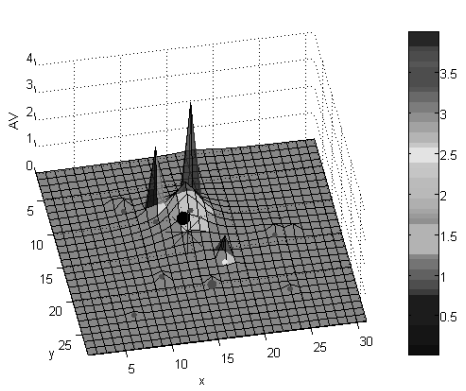


Figure 3. Example Output of the Cognitive Model of Visual Attention (Bosse et al., 2006).

4.2 Tactical Picture Compilation Simulator

The characterizations of different attentional states in relation with adaptive task allocation was investigated in another case study, namely one in the naval surface warfare domain. In Figure 4 a snapshot of the interface of the Tactical Picture Compilation (TPC) Simulator, used in this study,

[2]For more information see `http://www.tobii.se`.

[3]See `http://www.few.vu.nl/~pp/attention` for a complete animation.

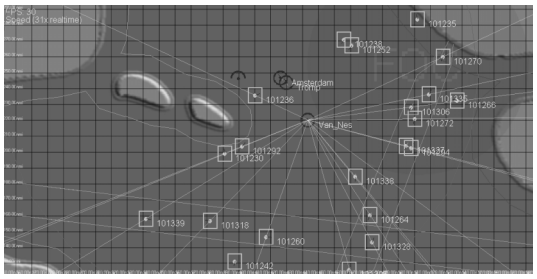


Figure 4. Tactical picture compilation simulation that was used to implement a dynamic task allocator based on a cognitive model of attention.

is shown. Similar to the case study in the previous section, this study was also conducted in an effort to augment the meta-cognitive capabilities of the user in naval operations in naval operations and to leverage cognitive models of attention. However, the present study focused on a more realistic domain: its goal was to establish an implicit work division between a naval officer and a supportive agent based on a cognitive model of the TPC task. TPC is a critical task in naval surface warfare. It is continuously performed by naval warfare officers during operations at sea. The main goal is to create an accurate tactical picture of the immediate surroundings of the ship using the ship's sensors. Changes in the tactical picture can occur either extremely slow or very rapidly depending on the traffic density at sea.

The application contained a TPC agent that supported the officer's interpretation of the behavior of ships in the area. The officer observes the behavior of the ships in the area over time in order to deduce if they have hostile intent or not. The attentional capacity of the officer may come under stress during transitional periods from low traffic densities to high traffic densities or during intense tactical action. During these periods of high cognitive workload, the officer is supported by the TPC agent. The question is how to determine the optimal division of workload without adding to the workload of the officer by letting him or her decide what the division should be. Instead, the main region of interest on the radar screen is determined by using a cognitive model of attention. The model is similar to the one used for the study mentioned in Section 4.1. It deduces the region of interest based on an interpretation of eye gaze behavior of the officer as well as information about the properties of the objects and spaces visible on the radar screen. The various symbols and lines are not equally visually salient and the model must correct for this. The regions that are not covered by the modeled attention of the officer are then assigned to the TPC agent for processing. The officer therefore *implicitly* communicates the meta-cognitive decision which radar tracks are desired to be handled by the TPC agent.

This implicit communication is based on the assumption that the user's cognitive performance in determining threat is better than that of the TPC agent. This means that no prescriptive variant of the TPC model was used to determine the work division between the TPC agent and the officer. A prescriptive model of the TPC task, as described in Section 3, may be used to inform the work division decision, by checking if the officer pays attention to the radar tracks that represent the greatest threat. If not, then the optimal work division may need to be re-assessed. The TPC agent will automatically be assigned those tracks that fall outside the scope of attention of the officer. In future versions of this application, a prescriptive model can be implemented to enhance performance in situations where threat is ambiguous or situations that contains multiple threats from different directions. In those situations, the most optimal work division may consist of the officer covering one threat and the TPC agent covering another, instead of just keeping track of nominal radar tracks. A prescriptive TPC model is then required to detect threats outside the scope of attention of the officer.

This form of decision support is implicit in the sense that there is no explicit communication between officer and agent about the decisions supported. Both the work division decision and task related decisions happen automatically. An interesting question is to determine whether the acceptance of this form of decision support by naval officers can be enhanced if the system communicates to the officer the reasons on which it bases its decisions. This might cause the officer to trust the system more, because the reasons on which decisions are made inspire confidence. On the other hand, situations might develop in which the officer generates expectations about the decisions made by the supporting agent.

This might lead to a form of paranoia in which the officer is distracted from the main task (TPC) because of the desire to check the decisions of the supporting agent.

5 Discussion

The Augmented Cognition International Society defines augmented cognition as "an emerging field of science that seeks to extend a user's abilities via computational technologies, which are explicitly designed to address bottlenecks, limitations, and biases in cognition and to improve decision making capabilities." The Society also formulated a goal "to develop computational methods and neurotech tools that can account for and accommodate information processing bottlenecks inherent in human-system interaction (e.g., limitations in attention, memory, learning, comprehension, visualization abilities, and decision making)." Augmented cognition is a wide area, that is applicable to various types of cognitive processes. As the area develops further, it may be useful to differentiate the field a bit more, for example, by distinguishing more specific classes of application.

In this paper, such a distinction is put forward: augmented cognition focusing on task content versus augmented cognition focusing on task coordination. As the latter is considered a form of meta-cognition, this suggests augmented meta-cognition as an interesting sub-area of augmented cognition. The paper discussed applications to the meta-cognition used for dynamic task allocation within this area. It has been pointed out how functioning of human-computer systems can be improved by incorporating augmented meta-cognition in them. Especially in tasks involving multiple stimuli that require fast responses, this concept is expected to provide a substantial gain in effectiveness of the combined system.

Acknowledgments

This research was partly funded by the Royal Netherlands Navy under program number V524.

References

Bainbridge, L. (1983). Ironies of automation. *Automatica*, 19:775–779.

Bosse, T., van Maanen, P.-P., and Treur, J. (2006). A cognitive model for visual attention and its application. In Nishida, T., editor, *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT-06)*, pages 255–262. IEEE Computer Society Press.

Campbell, G., Cannon-Bowers, J., Glenn, F., Zachary, W., Laughery, R., and Klein, G. (1997). Dynamic function allocation in the sc-21 manning initiative program. Technical report, Naval Air Warfare Center Training Systems Division, SC-21/ONRS&T Manning Affordability Initiative, Orlando.

Clamann, M. P., Wright, M. C., and Kaber, D. B. (2002). Comparison of performance effects of adaptive automation applied to various stages of human-machine system information processing. In *Proceedings of the 46th Annual Meeting of the Human Factors and Ergonomics Society*, pages 342–346.

Green, D. and Swets, J. (1966). *Signal detection: theory and psychophysics*. Wiley.

Horvitz, E., Pavel, M., and Schmorrow, D. D. (2001). *Foundations of Augmented Cognition*. National Academy of Sciences.

Inagaki, T. (2003). Adaptive automation: Sharing and trading of control. In Hollnagel, E., editor, *Handbook of Cognitive Task Design*, pages 147–169. LEA.

Kaber, D. B. and Endsley, M. R. (2004). The effects of level of automation and adaptive automation on human performance, situation awareness and workload in a dynamic control task. *Theoretical Issues in Ergonomics Science*, 5(2):113–153.

Maes, P. and Nardi, D. (1988). *Meta-level Architectures and Reflection*. North-Holland.

Schmorrow, D. D. and Kruse, A. A. (2004). Augmented cognition. *Berkshire Encyclopedia of Human-Computer Interaction*, pages 54–99.

Chapter 10

Simulation and Formal Analysis of Visual Attention

This chapter appeared as (Bosse et al., 2009f), which was partially based on (Bosse et al., 2006, 2007e,d). Also an extended abstract (Bosse et al., 2007c) appeared based on (Bosse et al., 2006).

Simulation and Formal Analysis of Visual Attention

Tibor Bosse*, Peter-Paul van Maanen*† and Jan Treur*

* Department of Artificial Intelligence, Vrije Universiteit Amsterdam
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
Email: {tbosse, treur}@cs.vu.nl
† Department of Cognitive Systems Engineering, TNO Human Factors
P.O. Box 23, 3769 ZG Soesterberg, The Netherlands
Email: peter-paul.vanmaanen@tno.nl

***Abstract*—In this paper a simulation model for visual attention is discussed and formally analyzed. The model is part of the design of an agent-based system that supports a naval officer in its task to compile a tactical picture of the situation in the field. A case study is described in which the model is used to simulate a human subject's attention. The formal analysis is based on temporal relational specifications for attentional states and for different stages of attentional processes. The model has been automatically verified against these specifications.**

***Index Terms*—Visual Attention, Ambient Intelligence, Cognitive Modeling, Simulation, Philosophical Foundations.**

1 Introduction

This paper presents a formal model of visual attention, which is part of an agent-based system that supports a user in his task (Bosse et al., 2006, 2007b,c). Being able to formally describe human attention opens the opportunity to run real-time simulations and apply formal analyses. These can be beneficial for a number of reasons. First of all, on a theoretical level, the attempt can be used to enhance the understanding of human attentional processes. On a more practical level, simulation and formal analysis of attention can be beneficial as well. For example, in the domain of naval warfare, a crucial but complex task is tactical picture compilation. In this task, the naval officer has to compile a tactical picture of the situation in the field, based upon the information he observes on a computer screen. Similar situations occur in other domains, for example, in the area of air traffic control. Since the environment in these situations is often complex and dynamic, the naval officer has to deal with a large number of tasks in parallel. Given the demanding nature of such tasks, in practice, the human is often supported by automated systems (agents) that take over part of these tasks or assist the execution of them. However, a problem is how to determine an appropriate work division between human and agent: due to the rapidly changing environment, such a work division cannot be fixed beforehand by the designers of the support systems (Bainbridge, 1983). This results in the need for agent systems that are able to dynamically and at run-time reallocate tasks between human and agent. For this purpose, two approaches exist, namely human-triggered and system-triggered dynamic task allocation (Campbell et al., 1997). In the former case, the user can decide up to what level the agent should assist him. However, especially in alarming situations, the user does not have enough time to think about task reallocations (Inagaki, 2003). In these situations it would be better if the system determines this. Hence a system-triggered dynamic task allocation is desirable. This is where simulation and formal analysis comes into play: it could enable agent systems to properly reallocate tasks by adapting to the human's attentional state and processes.

In order to obtain such a system-triggered dynamic task allocation, the model of visual attention introduced and formally analyzed in this paper is incorporated within a supporting agent. The idea is to use an estimation of the user's current allocation of attention to determine which subtasks the agent is best to pay attention to. In this case, it is the agent that adapts its support to the human's attentional state and processes. For instance, if one of a users

tasks is to identify a certain track on a computer screen, it is likely that the user desires support concerning this track, whereas the user probably does not desire support concerning those tracks that do not have to be identified. On the other hand, if a certain track that is not in the focus of the user's attention clearly requires attention, it is desirable that the supporting agent either takes over the task dealing with this track or notifies the user that he should pay attention to it, or a colleague of the user who can also perform the task. In this paper we assume that, if the users attention is allocated to certain objects, he is also committing himself to dealing with those in a correct manner. Given this assumption, the agent can adjust its support at runtime solely based on the dynamics of the modeled attention, i.e., it does not have to worry about any other problems once the tasks have been properly divided. This is a reasonable assumption, since attention is a prerequisite for conscious action (Baars, 1988) and the application is aimed at highly trained users, unlikely to make mistakes once their situation awareness is in order.

As mentioned above, this paper introduces a model of visual attention that can be incorporated within a supporting agent (Bosse et al., 2006, 2007b,c). In addition, the model is formally analyzed, by using the output of a simulation based on an implementation of the model and data from a case study. In this case study, a user executed a task abstracted from an Air Traffic Control (ATC) task. This ATC task was tailored to a naval radar track identification task, because this suited more the domain of this study. The gathered data from the case study, which is only used for demonstration purposes, consist of two types of information: dynamics of tracks on a radar scope; dynamics of the user's gaze. Based on this information, together with the knowledge of the rules of the task, the cognitive model estimates the distribution of attention over the different locations on the radar scope. Furthermore, based on the characteristics of this attention distribution over time, temporal properties are defined that indicate certain attentional subprocesses. These subprocesses are related to the different phases of information processing (LaBerge, 2002; Parasuraman, 1998; Pashler et al., 2001). The further discrimination of attention is juxtaposed to the assumption often made in literature that attention is a single and homogeneous concept (e.g., Itti and Koch, 2001; Theeuwes, 1994).

This paper is structured in the following manner. In Section 2 a brief introduction of the existing literature on visual attention is introduced. The sole purpose of this part is to help understand the choices made further on. Next, Section 3 shows a mathematical description of the cognitive model of attention. This description is quite straightforward and the main contribution of the present paper is in the application and the formal analysis of such models in adaptive agent-based systems that assist human users. One of such applications is illustrated in Section 4, which consists of a description of a case study and the corresponding simulation results after applying the model to the case study, using human data. Section 5 shows how the model can be further analyzed by verifying formal temporal relational specifications for attentional states and subprocesses. Section 6 is a discussion. At the end of this paper an appendix is included that consists of the source code of the implemented cognitive model.

2 Visual Attention

Visual attention has been a subject of study in many disciplines and this section is not intended to deliberate on all of these disciplines. It rather discusses a small but dominant part of the literature on attention, in order to bridge between relevant theory on the one hand and the application mentioned in the introduction on the other hand. This bridge helps to explain why certain choices are made later on in the paper.

In Psychology, a dominant view on attention distinguishes two types of attention: exogenous attention and endogenous attention (Theeuwes, 1994). The former stands for attention by means of triggers by (partially) unexpected inputs from the environment, i.e., bottom-up triggers, such as a fierce blow on a horn. The latter stands for attention by means of a slower trigger from within the subject, i.e., top-down triggers, such as searching a friend in a crowd. There are reasons to say that exogenous and endogenous attention are closely intertwined. A recent study (Pashler et al., 2001), for instance, shows that capture of exogenous attention occurs only if the

object that attracts attention has a property that a person is using to find a target.

Another relevant aspect of visual attention for modeling is the effect of so-called in-attentional blindness (Mack and Rock, 1998). This is the property that perception does not always result in attending to the important and unexpected events. Attention may also be a based on certain non-visual cognitive activities, such as having deep thoughts on past or future events. Because of the limited amount of attentional resources, this can result in a large blind spot for visual stimuli. Attention is therefore often distinguished in at least two types of attention, i.e., perceptual and decisional attention (Pashler et al., 2001). Some even propose a larger number of functionally different subprocesses of attention (LaBerge, 2002; Parasuraman, 1998). This gives rise to the idea that attention is more than of a single homogeneous type, which should be taken into account.

A third important discussion in the psychological literature relevant for computational models of attention addresses the distinction of two definitions of visual attention: the definition of visual attention as a division over space and the definition as a division over objects. The first definition is more traditional and involves continuous locations over $2D$ or $3D$ space. There are several space-based theories of attention, such as the filter theory (Broadbent, 1958), spotlight theory (Posner, 1980), and the zoom-lens theory (Eriksen and St. James, 1986). There are many more, but to mention and discuss them is beyond the scope of the paper. What they all have in common is that attention is subject to whatever is within a certain location in space. The object-based view of attention is more 'recent' and stresses that attention is allocated to (groups of) perceptual objects, rather than a continuous space (Duncan, 1984). These objects can have various properties, such as shape, speed, color, etc. Location, in that sense, is treated as just a special property of objects. Computationally this seems intriguing, but there are downsides of this view, which is treated below.

A fourth important and relevant discussion in psychology is sometimes called to be related to the what-where-distinction (Logan, 1996), and combines in some way the space- and object-based views of attention. So-called what-attention prepares a person that something will happen concerning a certain already visible object. Where-attention, on the other hand, prepares the sensory memory for further deliberation. This kind of preparation also happens when a person expects something to happen in a specific region in the search space or from a sensory input, but does not know what exactly will happen. One of the downsides of the above mentioned object-view is that it is difficult to translate such a kind of where-attention according to that view.

Apart from a more experimental interest, mainly in Psychology, there has also been a growing interest in the development and application of mathematical models of visual attention, mainly in Computer Science and AI (Itti and Koch, 2001). Such models are for instance used for enhancing encryption techniques in JPEG and MPEG standards (Chen et al., 2003). Another example of an application is usage of such models for making believable virtual humans in artificial environments (Kim et al., 2005). Basically one can distinguish two types of questions addressed within the literature on visual attention modeling:

- Given certain circumstances and behavior, to which attention distribution does this lead?
- Given certain circumstances and an attention distribution, to which behavior does this lead?

Models addressing the first question are for instance interesting for predicting to what features of an image a person is paying attention. Models addressing the second question are for instance interesting for generating realistic behavior for virtual characters. Answers to both questions help in the construction of cognitive models of visual attention.

To construct cognitive models of attention, several types of information can be used as an input. In general, the following three types of information are distinguished:

- Behavioral cues from the user. Information or cues about the behavior of the user are dependent on the current attentional state. The behavior is a result of it and this means that, for example, gaze duration, gaze frequency, gaze path, head pose, and task performance, are usable as input of the model.
- Properties of objects in the environment. This

type of information can lead to clues about when certain stimuli from the environment cause a user to attend to an object. Examples of such cues are features of objects, such as shape, texture, color, size, movement, direction, and centeredness. Note that this type of information addresses exogenous attention.

- Properties of the human attention mechanism. An example of this kind of information is knowledge about when a user pays attention to a speaker. For instance only when he expects or wants the speaker to speak, or when he has a certain commitment, interest, or goal, related to the speaker. An estimate of the users commitments, interests, or goals, can lead to an estimate of an attention distribution and therefore be used as an input of the model. Note that this type of information addresses endogenous attention.

The next section will demonstrate how the above types of information can be integrated into one executable model.

3 A Mathematical Model for Visual Attention

In this section the mathematical model for visual attention is presented. The proposed model is composed of formal rules that are related to the psychological concepts discussed in the previous section. In Section 3.1 a formal definition of attention is given, taking into account the distinction between the two possible informal definitions stated earlier. In Section 3.2 it is described how behavioral cues of the user are derived from gaze characteristics and are used to estimate attention. In Section 3.3 saliency maps are discussed shortly, that translate properties of objects in the environment to a probable attention demand. Saliency maps are not only related to exogenous but also to endogenous attention, since saliency is task related as well. Inattentional blindness is modeled by means of fixing a certain limited amount of total attention, which is managed by normalization, persistency, decay, and concentration processes, described in Sections 3.4, 3.5 and 3.6 respectively.

3.1 Attention Values, Objects and Spaces

As described in Section 2, there is a distinction between the definition of attention as a division over space and that as a division over objects. In this paper the first approach is used and it is assumed that one can have attention for multiple spaces at the same time. One of the reasons for using spaces instead of objects is that it is actually possible to pay attention to certain spaces that do not contain any objects (yet).

The model presented in this paper will define different (discrete) spaces, which each have a specific 'quantity' of attention. One argument for this choice is that certain spaces can contain more relevant information than others. This quantity of attention will be called the attention value. Division of attention is now defined as an instantiation of attention values AV for all attention spaces s. An attentional state is a division of attention at a certain moment in time. Mathematically, given the above, the following is expected to hold:

$$A(t) = \sum_{spaces\ s} AV(s,t) \qquad (1)$$

where $A(t)$ is the total amount of attention at a certain time t and $AV(s,t)$ is the attention value for attention space s at time t. In this study we define attention spaces to be 1×1 squares within an $M \times N$ grid. In principle it holds that the more attention spaces, the less attention value for each of those spaces. This is reasonable because there is a certain upper limit of total amount of working memory humans have. In the following sections the concept of attention value is further formalized.

3.2 Gaze

As discussed earlier, human behavior can be used to draw conclusions on a person's current attentional state. An important aspect of the visual attentional state is human gaze behavior. The gaze dynamics (saccades) are not random, but say something about what spaces have been attended to (Carpenter, 1988; Land and Furneaux, 1997). Since people often pay more attention to the center than to the periphery of their visual space, the relative distance of each space s to the gaze point (the center) is an important factor in determining the attention value of s. Mathematically this is modeled as follows:

$$AV_{new}(s,t) = \frac{AV_{pot}(s,t)}{1 + \alpha \cdot r(s,t)^2} \quad (2)$$

where $AV_{pot}(s,t)$ is the potential attention value of s at time point t. For now, the reader is advised to assume that $AV_{new}(s,t) = AV(s,t)$. The term $r(s,t)$ is taken as the Euclidian distance between the current gaze point and s at time point t (multiplied by an importance factor α which determines the relative impact of the distance to the gaze point on the attentional state):

$$r(s,t) = d_{eucl}(gaze(t), s) \quad (3)$$

Other ways for calculating attention degradation as a function of distance is for instance using a Gaussian approximation.

3.3 Saliency Maps

Still unspecified is how the potential attention value $AV_{pot}(s,t)$ is to be calculated. The main idea here is to use the properties of the space (i.e., of the types of objects present) at that time. These properties can be for instance features such as color, intensity, and orientation contrast, amount of movement (movement is relatively well visible in the periphery), etc. For each of such a feature a specific saliency map describes its potency of drawing attention (Chen et al., 2003; Itti and Koch, 2001; Itti et al., 1998). Because not all features are equally highlighting, an additional weight for every map is used. Formally the above can be depicted as:

$$AV_{pot}(s,t) = \sum_{maps\ M} M(s,t) \cdot w_M(s,t) \quad (4)$$

where for any feature there is a saliency map M, for which $M(s,t)$ is the unweighted potential attention value of s at time point t, and $w_M(s,t)$ is the weight for saliency map M, where $1 \leq M(s,t)$ and $0 \leq w_M(s,t) \leq 1$. The specific features used and the exact values for the weights depend on the application.

3.4 Normalization

The total amount of human attention is assumed to be limited. Therefore the attention value for each space s is limited due to the attention values of other attention spaces. This can be written down as follows:

$$AV_{norm}(s,t) = \frac{AV_{new}(s,t)}{\sum_{s'} AV_{new}(s',t)} \cdot A(t) \quad (5)$$

where $AV_{norm}(s,t)$ is called the normalized attention value for space s at time point t.

3.5 Persistency and Decay

On the one hand, visual attention is something that persists over time. If one has a look at a certain space at a certain time, it is probably not the case that the attention value of that space is lowered drastically the next moment (Theeuwes, 1994). This can be done by persistently keeping the model fed with input from the environment or the user, such as saliency and gaze, respectively. But, and this holds especially for gaze, the input is not persistent. Gaze is in general more dynamic than attention. Consider the following: reading this long sentence does not cause you to just pay attention to, and therefore comprehend, merely the characters you read, but instead, while your gaze follows specific positions in this sentence, you pay attention to whole parts of this sentence.

As a final observation, in reality it is impossible to keep one's attention to everything that one sees. In fact, given the above formulas, this will lead to increasingly low attention values (consider the formula in the previous section again).

Based on the above considerations a persistency and decay factor has been added to the model, which allows attention values to persist over time independently of the persistency of the input, but not completely: with a certain decay. Formally this can be described as follows:

$$\begin{aligned} AV(s,t) &= \lambda \cdot AV(s,t-1) \\ &\quad +(1-\lambda) \cdot AV_{norm}(s,t) \end{aligned} \quad (6)$$

where λ is the decay parameter that results in the decay of the attention value of s at time point $t-1$. Note that higher values for λ results in a higher persistency and lower decay and vice versa.

3.6 Concentration

In this document concentration is seen as the total amount of attention one can have. For instance if for all t, $A(t) = 1$, then the concentration is always the same, i.e., 1. But there may be a variance in concentration. Distractions by irrelevant stimuli can be the reason for that, or becoming tired. If the model needs to describe attention dynamics precisely and the task is sensitive for irrelevant distraction, one might consider non-fixed $A(t)$ values.

4 Case Study

Now that the model of visual attention has been explained, in this section a case study is briefly set out. The case study involves a human operator executing a naval officer-like task. For this case study, it is first explained how the data were obtained (Section 4.1). The data were then used as input for the simulation model (implemented in Matlab), which is described in detail in Section 4.2. In Section 4.3 the results of the case study are shown.

4.1 Task

The model of visual attention presented above was used in a simulation run based on 'real' data from a human participant executing a naval officer-like task. The software *Multitask* (Clamann et al., 2002) was altered in order to have it output the proper data as input for the model. This study did not yet deal with altering levels of automation (subject of Clamann et al.'s), and the software environment was momentarily only used for providing relevant data. Multitask was originally meant to be a low fidelity air traffic control (ATC) simulation. In this study it is considered to be an abstraction of the cognitive tasks concerning the compilation of the tactical picture. A snapshot of the task is shown in Figure 1.

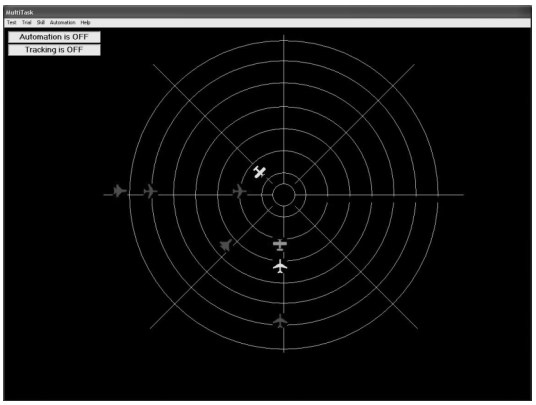


Figure 1. The interface of the experimental environment (Clamann et al., 2002).

In the case study the participant (controller) had to manage an air space by identifying aircrafts that all are approaching the center of a radarscope. The center contains a high value unit (HVU) and had to be protected. In order to do this, airplanes needed to be cleared and identified to be either hostile or friendly. Clearing contained six phases: 1) a red color indicated that the identity of the aircraft was still unknown, 2) flashing red indicated that the naval officer was establishing a connection link, 3) yellow indicated that the connection was established, 4) flashing yellow indicated that the aircraft was being cleared, 5) green indicated that either the aircraft was attacked when hostile or left alone when friendly or neutral, and finally 6) the target is removed from the radarscope when it reaches the center. Each phase consisted of a certain amount of time and to go from phase 1 to 2 and from phase 3 to 4 required the participant to click on the left and the right mouse button, respectively. Three different aircraft types were used: military, commercial, and private. Note here that the type did not determine anything about the hostility. The different types merely resulted in different intervals of speed of the aircrafts. All of the above were environmental stimuli that resulted in change of the participant's attention.

The data that were collected consisted of all locations, distances from the center, speeds, status of the aircrafts (which phase), and types. Additionally, data from a *Tobii x50 eye-tracker*[1] were extracted while the participant was executing the task. All data were retrieved several times per second. Together with the data from the experimental environment they were used as input for the simulation model described below.

[1]See `http://www.tobii.se` for more information.

4.2 Simulation Model

To obtain a simulation model, the mathematical model as shown in Section 3 has been implemented in Matlab. The behavior of the model can be summarized as follows. Every time step (100 ms), the following three steps are performed:

1) First, per location, the "current" attention level is calculated. The current attention level is the weighted sum of the values of the (possibly empty) tracks on that location, divided by $1+\alpha$ times the square of the distance between the attended location and the location of the gaze, according to the formula presented in Section 3.3.
2) Then, the attention level per location is normalized by multiplying the current attention level with the total amount of attention that the person can have and dividing this by the sum of the attention levels of all locations (also see Section 3.4).
3) Finally, per location, the "real" attention level is calculated by taking into account the history of the attention. Here a constant λ is used that indicates the decay, i.e., the impact of the history on the new attention level (compared to the impact of the current attention level), also see Section 3.5.

Before implementing the model in Matlab, first a simple prototype of the model (specified at a conceptual level) has been created, for testing purposes. To this end, the *LEADSTO language* Bosse et al. (2007a) has been used. This language is well suited for this purpose, because it allows models to be conceptual and executable at the same time. This language is based on direct temporal (e.g., causal) relationships of the following format: Let α and β be state properties of the form 'conjunction of atoms or negations of atoms', and e, f, g, h non-negative real numbers. In LEADSTO $\alpha \rightarrow\!\!\!\!\rightarrow_{e,f,g,h} \beta$ means:

> *If state property α holds for a certain time interval with duration g then after some delay (between e and f) state property β will hold for a certain time interval of length h.*

For more details of the LEADSTO language, see Bosse et al. (2007a).

Table I
CONSTANTS CORRESPONDING TO THE FORMULAE IN SECTION 3.

total duration of the simulation in time steps	500
highest x-coordinate	31
highest y-coordinate	28
$w_{stat}(s(x,y),t)$, weight factor of attribute status at space s and time point t	0.8 (for all s and t)
$w_{dist}(s(x,y),t)$, weight factor of attribute distance at space s and time point t	0.5 (for all s and t)
$w_{type}(s(x,y),t)$, weight factor of attribute type at space s and time point t	0.1 (for all s and t)
$w_{spd}(s(x,y),t)$, weight factor of attribute speed at space s and time point t	0.5 (for all s and t)
concentration $A(t)$, i.e., total amount of attention a person has at time point t	100 (for all t)
impact α of gaze on the current attention level	0.3
decay parameter λ, i.e., impact of history on the new attention level	0.8

The three steps described above can be represented in LEADSTO by the following causal relationships (also called Local Properties or LP's). Note that LP1, LP2 and LP3 correspond to the three steps described above. LP4 is used only to make sure that the real attention level becomes the old attention level after each round. First, some constants and sorts are introduced (which correspond to the parameters as used for the formulae as introduced in Section 3) in Table I.

LP1 Calculate Current Attention Level
Calculate the current attention level per location. The current attention level of a location is based on the values of the attributes of the (possibly empty) tracks on that location, and the distance between the location and the location of the gaze.

$\forall x1,x2,y1,y2$:COORDINATE $\forall v1,v2,v3,v4$:INTEGER $\forall tr$:TRACK is_at_location(tr, loc($x1$,$y1$)) $\wedge$ gaze_at_loc($x2$,$y2$) $\wedge$ has_value_for(tr, $v1$, status) $\wedge$ has_value_for(tr, $v2$, distance) $\wedge$ has_value_for(tr, $v3$, type) $\wedge$ has_value_for(tr, $v4$, speed) $\rightarrow\!\!\!\!\rightarrow_{0,0,1,1}$ has_current_attention_level(loc($x1$, $y1$), $(v1*w1+v2*w2+v3*w3+v4*w4)$ / (1 + α * $(x1-x2)$^2 + $(y1-y2)$^2))

LP2 Normalize Attention Level
Normalize the attention level per location by multiplying the current attention level with the total amount of attention, divided by the sum of the attention levels of all locations.

$\forall x1,y1$:COORDINATE $\forall v$:REAL
has_current_attention_level(loc($x1$,$y1$), v) $\wedge$
$s = \sum_{x2=1}^{\max\_x} [\ \sum_{y2=1}^{\max\_y}$ current_attention_level(loc($x2$,$y2$))]
$\rightarrow\!\!\!\!\rightarrow_{0,0,1,1}$ has_normalised_attention_level(loc(x1,y1), $v*a/s$)

LP3 Calculate Real Attention Level
Calculate the real attention level per location. The real attention level of a location is the sum of the old attention level times λ and the current (normalized) attention level times $1 - \lambda$.

$\forall x,y$:COORDINATE $\forall v1,v2$:REAL
has_normalised_attention_level(loc(x,y), $v1$) $\wedge$
has_old_attention_level(loc(x,y), $v2$) $\rightarrow\!\!\!\!\rightarrow_{0,0,1,1}$
has_real_attention_level(loc(x, y), λ*$v2$ + $(1 - \lambda)$*$v1$)

LP4 Determine Old Attention Level
After each round, the real attention level becomes the old attention level.

$\forall x,y$:COORDINATE $\forall v$:REAL
has_real_attention_level(loc(x,y), v) $\rightarrow\!\!\!\!\rightarrow_{\text{round}-2,\ \text{round}-2,1,1}$
has_old_attention_level(loc(x,y), v)

When this LEADSTO model turned out to show the expected behavior, it has been converted to an actual implementation in the mathematical environment Matlab. The Matlab code of the model can be found in the appendix.

4.3 Simulation Results

The results of applying the attention model to the input data described above are in the form of an animation, see. A screen-shot of this animation for one selected time point (i.e., time point 193) is shown in Figure 2. This figure indicates the distribution of attention over the grid at time point 193 (i.e., 19300 ms after the start of the task). The x- and y-axis denote the x- and y-coordinates of the grid, and the z-axis denotes the level of attention. As described earlier, the grid (which originally consists of 11760×10380 pixels) has been divided in a limited (31×28) number of locations. Besides the value at the z-axis, the color of the grid also denotes the level of attention: blue locations indicate that the location does not attract much attention, whereas green and (especially) red indicate that the location attracts more attention (see also the color bar at the right). In addition, the locations of all tracks are indicated in the figure by means of small "•" symbols. The colors of these symbols correspond to the colors of the tracks in the original task (i.e., red, yellow or green). Furthermore, the location of the gaze is indicated by a big blue "*" symbol, and a mouse click is indicated by a big black "●" symbol. Figure 2 clearly shows that at time point 193 there are two peaks of attention: at locations $(12, 10)$ and $(16, 9)$. Moreover, a mouse click is performed at location $(16, 9)$, and the gaze of the subject is also directed towards that location.

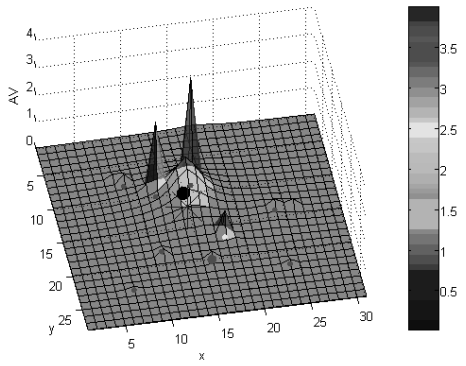


Figure 2. Attention distribution at time point 193.

5 Temporal Relational Specification and Verification

This section addresses formal analysis of the behavior of the simulation model. To this end, it is shown how (temporal) properties of states and processes concerning visual attention can be formally specified and verified against simulation traces. In particular, in Section 5.1, backward and forward temporal relational specifications for attentional states are discussed, and in Section 5.2 temporal relational specifications for different attentional subprocesses. In Section 5.3 it is shown how these formally specified temporal relations can

be automatically verified. Section 5.4 presents the results of the verification.

5.1 Temporal Relational Specification of Attentional States

Although the work reported in this paper focuses on a practical application context, also a formal analysis for the notion of attentional state is discussed. More specifically, *representation relations* for attentional states are identified and formalized. These are relationships between the occurrence of a specific mental state and circumstances that occurred in the past or will occur in the future. To express such representation relations, the *relational specification* approach from Philosophy of Mind is adopted. This approach indicates how the occurrence of a mental state property relates to properties of states 'distant in space and time' (Kim, 1996, pp. 200–202). For a relational specification for a mental state property p, two possibilities are considered:

1) relating the occurrence of p to events in the past (*backward temporal relation*)
2) relating the occurrence of p to behavior in the future (*forward temporal relation*)

Applied to the case of an attentional state, a *backward* temporal relational specification can be used to describe what brings about this state, for example, gaze direction and cues of objects that are observed; this corresponds to possibility 1 above. A *forward* temporal relational specification for attentional states describes what the effect of this state is in terms of behavior; this corresponds to possibility 2 above. Below it is shown how these approaches can be applied to attentional states.

To formally represent attentional states, a quantitative approach is taken. This allows us to consider certain levels of a mental state property p; in this case a mental state property is involved that is parameterized by a number: it has the form $p(a)$, where a is a number, denoting that p has level a (e.g., in the case considered, the amount a of attention for space s). By decay, levels decrease over time. For example, if λ is the decay rate (with $0<\lambda<1$), then at a next time point the remaining level may be λ*a, unless a new contribution is to be added to the level. Decisions for certain behavior may be based on a number of such state properties with different levels, taking into account their values; e.g., by determining the state property with the highest value, or the ones above a certain threshold (which may depend on the distribution of values over the different mental state properties, in the case considered here the attention levels for the different spaces).

For the *backward case*, the temporal relational specification involves a summation over different time points. Moreover, a decay rate λ with $0 < \lambda < 1$ is used. The *backward temporal relational specification* is expressed by:

> *There is an amount w of attention at space s, if and only if there is a history such that at time point 0 there was initatt(0,s) attention at s, and for each time point k from 0 to t an amount new attention newatt(k, s) is added for s, and w = initatt(0, s) * λ^t + $\sum_{k=0}^{t}$ newatt($t-k$,s) * λ^k, where newatt($t-k$, s) is the weighted sum at time $t-k-e$ of feature values for s divided by 1 plus the square of the distance of s to the gaze point and normalized for the set of spaces.*

Note that the logical 'if and only if' connective indicates that the expression on the past is both a necessary and sufficient condition on past circumstances for the attentional state to occur.

The *forward case* involves a behavioral choice that depends on the relative levels of the multiple mental state properties. This makes that at each choice point the temporal relational specification of the level of one mental state property is not independent of the level of the other mental state properties involved at the same choice point. Therefore it is only possible to provide a temporal relational specification for the combined mental state property. For the case considered, this means that it is not possible to consider only one space and the attention level for that space, but that the whole distribution of attention over all spaces has to be taken into account. The *forward temporal relational specification* is expressed in a bidirectional manner as follows:

> *If at time $t1$ the amount of attention at space s is above threshold h, then action*

is undertaken for s at some time $t2 \geq t1$ with $t2 \leq t1 + e$.

and:

If at some time $t2$ an action is undertaken for space s for track 1, then at some time $t1$ with $t2 - e \leq t1 \leq t2$ the amount of attention at space s was above threshold h.

Note that this statement expresses sufficient and necessary conditions for the attentional state to occur: the first clause is the necessary condition, and the second clause the sufficient condition for future circumstances. The threshold h can be determined, for example, as a value such that for 5% of the spaces the attention is above h and for the other spaces it is below h, or such that only three spaces exist with attention value above h and the rest under h.

5.2 Temporal Relational Specification of Attentional Sub-processes

In the previous subsection, temporal relational specifications for attentional states have been defined. In recent years, an increasing amount of work is aimed at identifying *different types* of attention, and focuses not on attentional states, but on *subprocesses* of attention. For example, many researchers distinguish at least two types of attention, i.e., perceptual and decisional attention(Pashler et al., 2001). Some others even propose a larger number of functionally different subprocesses of attention (LaBerge, 2002; Parasuraman, 1998). Following these ideas, this section provides a (temporal) differentiation of an attentional process into a number of different types of subprocesses. To differentiate the process into subprocesses, a cycle *sense–examine–decide–prepare and execute action–assess action effect* is used. It is discussed how different types of attention within these phases can be distinguished and defined by temporal specifications.

5.2.1 Attention allocation: This is a subprocess in which attention of a subject is drawn to an object by certain exogenous (stimuli from the environment) and endogenous (e.g., goals, expectations) factors (see e.g., Theeuwes, 1994). At the end of such an 'attention catching' process an attentional state for this object is reached in which gaze and internal focus are directed to this object. The informal temporal specification of this *attention allocation process* is as follows:

From time $t1$ to $t2$ attention has been allocated object O iff at $t1$ a combination of external and internal triggers related to object O occurs, and at $t2$ the mind focus and gaze are just directed to object O.

Note that in this paper validation only takes place with respect to gaze and not to mind focus, as the empirical data used have no reference to internal states.

5.2.2 Examinational Attention: Within this subprocess, attention is shared between or divided over a number of different objects. Attention allocation is switched between these objects, for example, visible in the changing gaze. The informal temporal specification of this *examinational attentional process* is as follows:

During the time interval from $t1$ to $t2$ examinational attention occurs iff from $t1$ to $t2$, for a number of different objects, attention is allocated alternatively to these objects.

5.2.3 Decision Making Attention: A next subprocess distinguished is one in which a decision is made on which object to select for an action on a certain object to be undertaken. Such a *decision making attentional process* may have a more inner-directed or introspective character, as the subject is concentrating on an internal mental process to reach a decision. Temporal specification of this attentional subprocess involves a criterion for the decision, which is based on the relevance of the choice made; it is informally defined as follows:

During the time interval from $t1$ to $t2$ decision making attention occurs iff at $t2$ attention is allocated to an object, from which the relevance is higher than a certain threshold.

5.2.4 Action Preparation and Execution Attention: Once a decision has been made for an action, an *action preparation and execution attentional process* occurs in which the subject concentrates on the object, but this time on the aspects relevant for action execution. The informal temporal specification is as follows:

> *During the time interval from $t1$ to $t2$ attention on action preparation and execution occurs iff from $t1$ to $t2$ the mind focus and gaze is on an object O and at $t2$ an action a is performed for this object O.*

5.2.5 Action Assessment Attention: Finally, after an action has been executed, a retrospective *action assessment attentional process* occurs in which the subject evaluates the outcome of the action. Here the subject focuses on aspects related to goal and effect of the action. For example, Wegner (2002) investigates such a process in relation to the experience of conscious will and ownership of action. The informal temporal specification of this attentional process is as follows:

> *During the time interval from $t1$ to $t2$ action assessment attention occurs iff at $t1$ an action a is performed for this an object O and from $t1$ to $t2$ the mind focus and gaze is on this object O and from $t2$ they are not on O.*

5.3 Formal Specification and Analysis

The temporal representation relations introduced in the previous sections are expected to apply to different parts of attentional processes. In order to verify in detail which of these specifications holds for which part of a given process, automated support is needed. This subsection explains how the representation relations introduced above can be automatically verified against the simulation runs presented in Section 4.3.

In order to analyse the results of the simulation model in detail, they first are converted into formally specified *traces* (i.e., sequences of events over time). Moreover, the temporal relational specifications discussed above are logically formalized in the language TTL (Bosse et al., 2009). This predicate logical language supports formal specification and analysis of dynamic expressions, covering both qualitative and quantitative aspects. TTL is built on atoms referring to states, time points and traces. Traces are time-indexed sequences of states. Here a *state* S is described by a truth assignment to the set of basic *state properties* (ground atoms) expressed using a state ontology Ont; i.e., $S : At(Ont) \rightarrow \{true, false\}$. A state ontology Ont is formally specified as a sets of sorts, objects in sort, and functions and relations over sorts. The set of all possible states for state ontology Ont is denoted by States(Ont). A trace γ is an assignment of states to time points; i.e., γ: TIME $\rightarrow$ States(Ont). To represent the empirical data of the case study described in Section 4, a state ontology based on the relations in Table II has been used.

Note that in the last four relations in Table I, v is an integer between 0 and 10. The idea is that, the higher the value of v, the more salient the corresponding track (aircraft) is for within this task. For example, a red track is more salient than a yellow track (since red tracks need to be clicked on more often before they are cleared), but a yellow track is assumed to be more salient than a flashing red track (since it is not possible to click on flashing tracks; one has to wait until they stop flashing). Based on the above state ontology, states are created by filling in the relevant values for the state atoms at a particular time point. Traces are built up as time-indexed sequences of these states. An example of (part of) a trace that resulted from the experiment is visualized in Figure 3. Each time unit in this figure corresponds to 100 ms in real time.

In addition to the above, *dynamic properties* are temporal statements that can be formulated with respect to traces based on the state ontology Ont in the following manner. Given a trace γ over state ontology Ont, the state in γ at time point t is denoted by state(γ, t). These states can be related to state properties via the formally defined satisfaction relation denoted by the infix predicate $\models$, comparable to the holds-predicate in the Situation Calculus: state$(\gamma, t) \models p$ denotes that state property p holds in trace γ at time t.

Based on these statements, dynamic properties (about representation relations) can be formulated

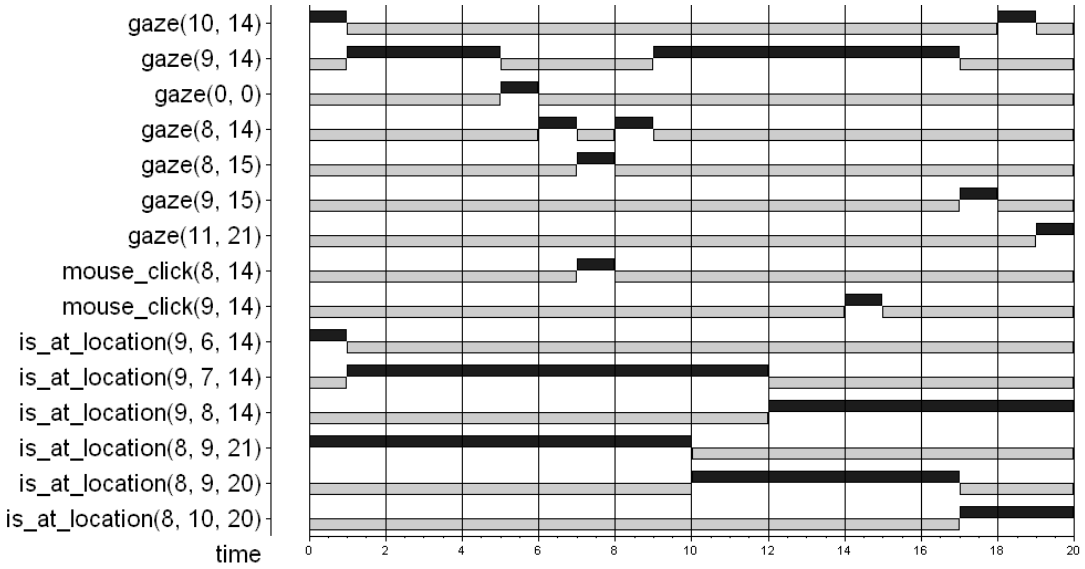


Figure 3. Visualization of (part of) the empirical trace on the interval $[65, 85)$. The vertical axis depicts atoms that are either true or false. This is indicated, respectively, by dark or light boxes on the horizontal axis, in units of 10 ms.

in a formal manner in a sorted first-order predicate logic, using quantifiers over time and traces and the usual first-order logical connectives such as $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\forall$ and $\exists$. A special software environment has been developed for TTL, featuring both a Property Editor for building and editing TTL properties and a Checking Tool that enables formal verification of such properties against a set of (simulated or empirical) traces. An example of a relevant dynamic property expressed in TTL is the following:

GP1 (Mouse Click implies High Attention Level Area)
For all time points t, if a mouse click is performed at location (x, y), then at e time points before t, within a range of 2 locations from (x, y), there was a location with an attention level that was at least h, where h is a certain threshold that can be determined as explained in the previous section. Formalization:

$\forall\ t$:$T\ \forall\ x$,y:COORDINATE
[state(γ,t) $\models$ mouse_click $(x, y) \Rightarrow$
high_attention_level_nearby(γ, $t - e, x, y$)]

Here, high_attention_level_nearby is an abbreviation, which is defined as follows:

high_attention_level_nearby(γ:TRACE, t:TIME,
x, y :COORDINATE) $\equiv$
$\exists p,q$: COORDINATE, $\exists i$:REAL
state(γ,t) $\models$ has_attention_level(p,q,i) &
$x - 2 \leq p \leq x + 2 \& y - 2 \leq q \leq y + 2 \& i > h$

Note that this property is a refinement of the forward temporal relational specification defined in Section 5.1. Roughly spoken, it states that for every location that the user clicks on, some time before (e time points) he had a certain level of attention. The decision to allow a certain error (see GP1: instead of demanding that there was a high attention level at the exact location of the mouse click, this is also allowed at a nearby location within the surrounding area) was made in order to handle noise in the retrieved empirical data. Usually, the precise coordinates of the mouse clicks do not correspond exactly to the coordinates of the tracks and the gaze data. This is due to two reasons:

1) a certain degree of inaccuracy of the eye tracker, and

2) people often do not click exactly on the center of a track.

Table II
STATE ONTOLOGY USED TO REPRESENT THE DATA.

gaze(x:COORD, y:COORD)	subject's gaze is currently directed at location (x, y)
is_at_location(i:TRACK_NR, x:COORD, y:COORD)	track (aircraft) with number i is currently at location (x, y)
mouse_click(x:COORD, y:COORD)	subject is clicking with the mouse on location (x, y)
has_attention_level(x:COORD, y:COORD, v:REAL)	location (x, y) currently has attention level v (according to the simulation model)
has_status(i:TRACK_NR, v:INT)	track i has status v; e.g., 'red'[a]
has_distance(i:TRACK_NR, v:INT)	distance between track i and the center of the screen is v[b]
has_type(i:TRACK_NR, v:INT)	track i has type v; e.g., 'military'[c]
has_speed(i:TRACK_NR, v:INT)	speed of track i is v[d]

[a] Here, 9 = "red", 8 = "yellow", 5 = "flashing red", 4 = "flashing yellow", 3 = "green", and 1 means that the track is currently not active.
[b] This v is calculated using the formula $v = 10 - (d/550)$, where d (which is a number between 0 and 5500) is the actual distance in pixels from the centre of the screen; $v = 1$ indicates that the track is currently not active.
[c] Here 8 = "military plane", 6 = "commercial plane", 4 = "private plane", and 1 means that the track is currently not active.
[d] The variable v is calculated using the formula $v = s/100$, where d (which is a number between 100 and 1000) is the actual speed (in pixels per second). Furthermore, $v = 1$ indicates that the track is currently not active.

The approach used is able to deal with such imprecision.

Besides GP1, also the temporal relations for attentional subprocesses introduced in Section 5.2 have been formalized, as shown below. To this end, first some useful help-predicates are defined:

gaze_near_track(γ:TRACE, c:TRACK, $t1$:TIME) $\equiv$
$\exists x1, y1, x2, y2$:COORDINATE
state(γ, $t1$) gaze($x1$, $y1$) &
state(γ, $t1$) is_at_location(c, $x2$, $y2$) &
$\mid x2 - x1 \mid \leq 1 \& \mid y2 - y1 \mid \leq 1$

mouseclick_near_track(γ:TRACE, c:TRACK, $t1$:TIME) $\equiv$
$\exists x1, y1, x2, y2$:COORDINATE
state(γ, $t1$) mouse_click($x1$, $y1$) &
state(γ, $t1$) is_at_location(c, $x2$, $y2$) &
$\mid x2 - x1 \mid \leq 1 \& \mid y2 - y1 \mid \leq 1$

action_execution(γ:TRACE, c:TRACK, $t2$:TIME) $\equiv$
mouseclick_near_track(γ, c, $t2$) &
$\exists t1$:TIME $t1 < t2$ & $\forall t3$:TIME $[t1 \leq t3 \leq t2 \Rightarrow$
gaze_near_track(γ, c, $t3$)]

The reason for using gaze_near_track instead of something like gaze_at_track is again that a certain error is allowed to handle noise. Based on these intermediate predicates, the five types of attentional (sub)processes as described earlier are presented below, both in semi-formal and in formal (TTL) notation:

GP2A (Allocation of attention)
From time $t1$ to $t2$ attention has been allocated to track c iff at $t2$ the gaze is directed to track c and between $t1$ and $t2$ the gaze has not been directed to any track.

has_attention_allocated_during(γ:TRACE, c:TRACK, $t1$, $t2$:TIME) $\equiv t1 < t2$ & gaze_near_track(γ, c, $t2$) &
$\forall t3$:TIME, $c1$:TRACK
$[t1 \leq t3 < t2 \Rightarrow \neg$ gaze_near_track(γ, $c1$, $t3$)]

GP2B (Examinational attention)
During the time interval from $t1$ to $t2$ examinational attention occurs iff at least two different tracks $c1$ and $c2$ exist to which attention is allocated during the interval from $t1$ to $t2$ (between $t3$ and $t4$ and between $t5$ and $t6$, respectively).

has_examinational_attention_during(γ:TRACE, $t1$, $t2$:TIME)
$\equiv \exists t3, t4, t5, t6$:TIME $\exists c1, c2$:TRACK
$t1 \leq t3 \leq t2 \& t1 \leq t4 \leq t2 \& t1 \leq t5 \leq t2 \& t1 \leq t6 \leq t2$ &
$c1 \neq c2$ & has_attention_allocated_during(γ, $c1, t3, t4$) &
has_attention_allocated_during(γ, $c2, t5, t6$)

GP2C (Attention on decision making and action selection)
During the time interval from $t1$ to $t2$ decision making attention for c occurs iff from $t1$ to $t2$ attention is allocated to a track c, for which the saliency at time point $t1$ (based on features type, distance, color and speed) is higher than a certain threshold th.

has_attention_on_action_selection_during(γ:TRACE, c:TRACK, $t1, t2$:TIME, th:INTEGER) $\equiv$
$t1 \leq t2$ & $\exists p1, p2, p3, p4$:VALUE $\forall t3$
[$t1 \leq t3 \leq t2 \Rightarrow$
state(γ, $t3$) has_type(c, $p1$) $\wedge$ has_distance(c, $p2$) $\wedge$
has_colour(c, $p3$) $\wedge$ has_speed(c, $p4$)] &
$(0.1 * p1 + 0.5 * p2 + 0.8 * p3 + 0.5 * p4)/1.9 > th$ &
has_attention_allocated_during(γ, c, $t1$, $t2$)

GP2D (Attention on action preparation and execution)
During the time interval from $t1$ to $t2$ attention on action preparation and execution for c occurs iff from some $t4$ to $t1$ attention on decision making and action selection for c occurred and from some $t3$ to $t2$ attention on the execution of an action on c occurs.

has_attention_on_action_prep_and_execution_during(
γ:TRACE, c:TRACK, $t1$, $t2$:TIME, th:INTEGER) $\equiv$
$t1 \leq t2$&$\exists t3$:TIME [$t3 \leq t1$ &
has_attention_on_action_selection_during (γ, c, $t3$, $t1$, th)]
& $\forall t4$:TIME $t1 \leq t4 \leq t2 \Rightarrow$
gaze_near_track(γ, c, $t4$) &
action_execution(γ, c, $t2$)

GP2E (Attention on action assessment)
During the time interval from $t1$ to $t2$ action assessment attention for c occurs iff at $t1$ an action on c has been performed and from $t1$ to $t2$ the gaze is on c and at $t2$ the gaze is not at c anymore.

has_attention_on_action_assessment_during(γ:TRACE,
c:TRACK, $t1$,$t2$:TIME) $\equiv$
[$t1 \leq t2$ & action_execution(γ, c, $t1$) &
$\neg$ gaze_near_track(γ, c, $t2$) &
$\forall t3$:TIME [$t1 \leq t3 < t2 \Rightarrow$ gaze_near_track(γ, c, $t3$)]

All the above TTL properties can be checked using the TTL Checking Tool (see next section). An example of how one could check such a property for certain parameters is the following:

check_action_selection $\equiv$
$\forall \gamma$:TRACE $\exists t1, t2$:TIME $\exists c$:TRACK
has_attention_on_action_selection_during (γ, c, $t1$, $t2$, 5)

This property states that the phase of decision making and action selection holds for track c, from time point $t1$ to time point $t2$, with a threshold of 5, for all loaded traces. This property either holds or does not. If so, the first instantiation of satisfying parameters are retrieved.

5.4 Analysis Results

In order to check automatically whether (and when) the above properties are satisfied by the empirical traces, the *TTL Checking Tool* (Bosse et al., 2009) has been used. This software takes a set of traces and a TTL property as input, and checks whether the property holds for the traces.

Using the TTL Checking Tool, property GP1 has been automatically verified against the traces that resulted from the case study. For these checks, e was set to 5 (i.e., 500 ms, which by experimentation turned out a reasonable reaction time for the current task), and h was set to 0.3 (which was chosen according to the 5%-criterion, see Section 5.1). Under these parameter settings, all checks succeeded. Although this is no exhaustive verification, this is an encouraging result: it shows that the subject always clicks on locations for which the model predicted a high attention level.

In addition to this, also property GP2A to GP2E turned out to hold for the formal traces, at least given the right parameter settings. This confirms the hypothesis that a temporal differentiation of a number of attentional subprocesses can be found in empirical data, namely those that are defined in terms of the above properties.

In addition to stating whether TTL properties hold, the Checking Tool also provides useful feedback about the exact instantiations of variables for which they hold. For example, suppose that the property check_action_selection holds for a certain trace, the checker will return specific values for time point $t1$ and $t2$ and for track c for which that property holds. This approach has been used to identify certain instances of attentional processes in the empirical traces that resulted from the experiment described earlier.

To illustrate this idea, Figure 3 shows part of such a trace.[2] For this trace, the five properties as

[2]Due to space limitations, in Figure 3 a mere selection of atoms has been made from the actual empirical trace, i.e., the time interval $[65, 85)$.

mentioned earlier hold for the following parameter values[3]:

- has_attention_allocated_during holds for track $c = 9$, for time points $t1 = 0$ and $t2 = 6$
- has_examinational_attention_during holds for tracks $c1 = 8$ and $c2 = 9$, for time points $t1 = 0$ and $t2 = 19$, because has_attention_allocated_during holds for track $c = 9$, for time points $t1 = 0$ and $t2 = 6$ *and* for track $c1 = 8$, for time points $t1 = 18$ to $t2 = 19$ (note that for $t = 17$ the gaze is still on track $c = 9$)
- has_attention_on_action_selection_during holds for track $c = 9$, for time points $t1 = 1$ and $t2 = 6$, and threshold value $th = 4$. This is due to the fact that has_type(9, 4), has_distance(9, 3), has_colour(9, 5), and has_speed(9, 4), not shown in Figure 3, result in a combined saliency above $th = 4$, for this time period
- has_attention_on_action_prep_and_execution_during holds for track $c = 9$, for time points $t1 = 6$ and $t2 = 7$, because has_attention_on_action_selection_during holds for track $c = 9$, for time points 1 to 6, and action_execution holds at $t2 = 7$, and the gaze is near track $c = 9$ at time point 7
- has_attention_on_action_assessment_during holds for track $c = 9$, for time points $t1 = 7$ and $t2 = 9$ (note that after $t2 = 9$ the gaze is not on $c = 9$ anymore)

Furthermore, the *TTL Checking Tool* enables additional analyses, such as counting the number of times a property holds for a given trace, using a built-in operator for summation. Using this mechanism, one can calculate that the property has_attention_allocated_during holds three times for track 1, four times for track 7, one time for track 8, and two times for track 9, in the time interval $[0, 100)$ of the empirical trace. A similar calculation shows that has_attention_on_action_prep_and_execution_during holds only once in this time interval, namely for track 9. For all other combinations of tracks and time intervals, the property does not hold. Comparison between such counts can be used to, for instance, indicate different task progresses or workload differences.

All of the formalized dynamic properties shown above have been successfully checked, given reasonable parameter instantiations, against the traces. Although an extensive investigation of the results is beyond the scope of this article, we hereby demonstrate the benefits of automated checks to investigate attentional processes. As mentioned above, the checks cannot be seen as an exhaustive validation, but they contribute to a detailed formal analysis of the simulation model. The main contribution of such an analysis is that it allows the user to distinguish different attentional states and subprocesses, which can be compared with the expected behavior.

6 Discussion

This paper presents a cognitive model as a component of a socially intelligent supporting agent (Dautenhahn, 2000). The component allows the agent to adapt to the need for support of its user, for example a naval officer and his or her task to compile a tactical picture. Given two types of input, i.e., user- and context-input, the implemented cognitive model is able to estimate the visual attention distribution within a $2D$-space. The user-input was retrieved by an eye-tracker, and the context-input by means of the output of the software for an Air Traffic Control task (ATC), tailored to a naval radar track identification task. The first consists of the (x, y)-coordinates of the gaze of the user over time. The latter consists of the variables speed, distance to the center, type of plane, and status of the plane. In a case study, the model was used to estimate the attention of a human user that executes the task mentioned above. The model was specifically tailored to domain-dependent properties retrieved from a task environment; nevertheless the method presented remains generic enough to be easily applied to other domains and task environments.

Although the work reported in this paper focuses on a practical application context, as a main contribution, also a formal analysis was given for attentional states and processes. To describe mental states of agents in general, the concept of *representational*

[3]Note that the maximum intervals are given for which each property holds. So, the fact that has_attention_allocated_during holds (for track $c = 9$) between time points 0 and 6, obviously implies that it also holds between time point 2 and 4. However, it does not hold for all time points after time point 6.

content is often applied, as described in the literature on Cognitive Science and Philosophy of Mind (e.g., Bickhard, 1993; Jacob, 1997; Jonker and Treur, 2003), (Kim, 1996, pp. 191–193, 200–202)[4]. In this paper this perspective first was applied to attentional states. The general idea is that the occurrence of the internal (mental) state property p at a specific point in time is related (by a *representation relation*) to the occurrence of other state properties, at the same or at different time points. Such a representation relation, when formally specified, describes in a precise and logically founded manner how the internal state property p relates to events in the past and future of the agent. To define a representation relation, the *causal-correlational approach* is often discussed in the literature in Philosophy of Mind. For example, the presence of a horse in the field has a causal relation to the occurrence of the mental state property representing this horse. This approach has some limitations (Jacob, 1997; Kim, 1996). Two approaches that are considered to be more generally applicable are the *interactivist approach* (Bickhard, 1993; Jonker and Treur, 2003) and the *relational specification approach* (Kim, 1996). In this paper the latter approach was adopted and formalized, as it provides the flexibility and expressiveness that is required to address issues as discussed below.

Fundamental issues that were encountered in the context of this work are (1) how to handle decay of a mental state property, (2) how to handle reference to a history of inputs, and (3) how to handle a behavioral choice that depends on a number of mental state properties. To address these, leveled mental state properties were used, parameterized by numbers. Decay was modeled by a kind of interest rate. Backward temporal relational specifications for attentional states were defined based on histories of contributions to attention, taking into account the interest rate. Forward temporal relational specifications for attentional states were defined taking into account combinations of multiple parameterized mental state properties, relating to the alternatives for behavioral choices. In addition, it has been shown how the notion of temporal relational specification can also be used to define and formalize different attentional subprocesses that play a role in the sense-reason-act cycle.

The temporal relational specifications have been formalized in the predicate logical language TTL. Using the *TTL Checking Tool*, they have been automatically verified against the traces that resulted from the case study. Under reasonable parameter settings, these checks turned out to succeed, which provides an indication that the attention model behaves as desired, and allows the user to get more insight into the dynamics of attentional processes. The approach used is able to handle imprecision in the data.

This paper focused on formal analysis; although in this formal analysis also empirical data were involved, a more systematic validation of the models put forward in the intended application context will be addressed as a next step. Future studies will address the use of the attention estimates for dynamically allocating tasks as a means for assisting naval officers. To determine (in a dynamical manner) an appropriate cooperation and work division between user and system, it has a high value for the quality of the interaction and cooperation between user and system, if the system has information about the particular attentional state or process a user is in. For example, in case the user is already allocated to some task, it may be better to leave that task for him or her, and allocate tasks to the system for which there is less or no commitment from the user (yet). A threshold can facilitate a binary decision mechanism that decides whether or not a task should be supported. Open questions are related to modeling both endogenous and exogenous triggers and their relation in one model. One important element missing is for example expectation as an endogenous trigger (Castelfranchi and Lorini, 2003; Martinho and Paiva, 2006). Finally, the attention model may be improved and refined by incorporating more attributes within the saliency maps, for example based on literature (e.g., Itti and Koch, 2001; Itti et al., 1998; Sun, 2003).

[4] A more exhaustive discussion of this theme from the philosophical perspective is beyond the scope of this paper.

ACKNOWLEDGMENTS

This research was partly funded by the Royal Netherlands Navy (program number V524).

REFERENCES

Baars, B. (1988). *A cognitive theory of consciousness*. Cambridge University Press, London.

Bainbridge, L. (1983). Ironies of automation. *Automatica*, 19:775–779.

Bickhard, M. H. (1993). Representational content in humans and machines. *Journal of Experimental and Theoretical Artificial Intelligence*, 5:285–333.

Bosse, T., Jonker, C. M., van der Meij, L., Sharpanskykh, A., and Treur, J. (2009). Specification and verification of dynamics in agent models. *International Journal of Cooperative Information Systems*, 18:167–193.

Bosse, T., Jonker, C. M., van der Meij, L., and Treur, J. (2007a). LEADSTO: a Language and Environment for Analysis of Dynamics by SimulaTiOn. *International Journal of Artificial Intelligence Tools*, 16(3):435–464.

Bosse, T., van Maanen, P.-P., and Treur, J. (2006). A cognitive model for visual attention and its application. In Nishida, T., editor, *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT-06)*, pages 255–262. IEEE Computer Society Press.

Bosse, T., van Maanen, P.-P., and Treur, J. (2007b). Simulation and formal analysis of visual attention in cognitive systems. In *Proceedings of the Fourth International Workshop on Attention in Cognitive Systems (WAPCV'07)*. Published as: L. Paletta, E. Rome (Eds.), Attention in Cognitive Systems, Lecture Notes in AI, Springer Verlag, 2007.

Bosse, T., van Maanen, P.-P., and Treur, J. (2007c). Temporal differentiation of attentional processes. In Vosniadou, S. and Kayser, D., editors, *Proceedings of the Second European Cognitive Science Conference (EuroCogSci'07)*, pages 842–847. IEEE Computer Society Press.

Broadbent, D. E. (1958). *Perception and Communication*. Pergamon Press, London.

Campbell, G., Cannon-Bowers, J., Glenn, F., Zachary, W., Laughery, R., and Klein, G. (1997). Dynamic function allocation in the sc-21 manning initiative program. Technical report, Naval Air Warfare Center Training Systems Division, SC-21/ONRS&T Manning Affordability Initiative, Orlando.

Carpenter, R. H. S. (1988). *Movements of the Eyes*. Pion, London.

Castelfranchi, C. and Lorini, E. (2003). Cognitive anatomy and functions of expectations. In *Proceedings of IJCAI 03 Workshop on Cognitive modeling of agents and multi-agent interaction*.

Chen, L., Xie, X., Fan, X., Ma, W., Zhang, H., and Zhou, H. (2003). A visual attention model for adapting images on small displays. *ACM Multimedia Systems Journal*.

Clamann, M. P., Wright, M. C., and Kaber, D. B. (2002). Comparison of performance effects of adaptive automation applied to various stages of human-machine system information processing. In *Proceedings of the 46th Annual Meeting of the Human Factors and Ergonomics Society*, pages 342–346.

Dautenhahn, K. (2000). *Human Cognition and Social Agent Technology*. John Benjamins Publishing Company.

Duncan, J. (1984). Selective attention and the organization of visual information. *Journal of Experimental Psychology*, 113:501–517.

Eriksen, C. and St. James, J. (1986). Visual attention within and around the field of focal attention: A zoom lens model. *Perception and Psychophysics*, 40(4):225–240.

Inagaki, T. (2003). Adaptive automation: Sharing and trading of control. In Hollnagel, E., editor, *Handbook of Cognitive Task Design*, pages 147–169. LEA.

Itti, L. and Koch, C. (2001). Computational modeling of visual attention. *Nature Reviews Neuroscience*, 2(3):194–203.

Itti, L., Koch, U., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:1254–1259.

Jacob, P. (1997). *What Minds Can Do: Intentionality in a Non-Intentional World*. Cambridge University Press.

Jonker, C. and Treur, J. (2003). A temporal-interactivist perspective on the dynamics of mental states. *Cognitive Systems Research Journal*, 4:137–155.

Kim, J. (1996). *Philosophy of Mind*. Westview Press.

Kim, Y., van Velsen, M., and Hill Jr., R. W. (2005).

Modeling dynamic perceptual attention in complex virtual environments. In Th. Panayiotopoulos, e. a., editor, *Proceedings of the Intelligent Virtual Agents, 5th International Working Conference (IVA 2005)*, pages 266–277.

LaBerge, D. (2002). Attentional control: brief and prolonged. *Psychological Research*, 66:230–233.

Land, M. F. and Furneaux, S. (1997). The knowledge base of the oculomotor system. *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, 352:1231–39.

Logan, G. D. (1996). The code theory of visual attention: an integration of space-based and object-based attention. *Psychol. Rev.*, 103:603649.

Mack, A. and Rock, I. (1998). Inattentional blindness: Perception without attention. In Wright, R. D., editor, *Visual Attention*, chapter 3, pages 55–76. MIT Press, Cambridge, MA.

Martinho, C. and Paiva, A. (2006). Using anticipation to create believable behaviour. In *Proceedings of AAAI06*.

Parasuraman, R. (1998). *The attentive brain*. MIT Press, Cambridge, MA.

Pashler, H., Johnson, J., and Ruthruff, E. (2001). Attention and performance. *Ann. Rev. Psych.*, 52:629–51.

Posner, M. I. (1980). Orienting of attention. *Quarterly Journal of Experimental Psychology*, 32:325.

Sun, Y. (2003). *Hierarchical Object-Based Visual Attention for Machine Vision*. PhD thesis, University of Edinburgh.

Theeuwes, J. (1994). Endogenous and exogenous control of visual selection. *Perception*, 23:429–440.

Wegner, D. (2002). *The illusion of conscious will*. MIT Press, Cambridge, MA.

APPENDIX

This appendix contains the *Matlab* code of the simulation model presented in Section 4.2. The complete model consists of four separate modules:

1) The *Pre-Processing Module*, needed to convert the data resulting from the task described in Section 4.1 and the data from the eye-tracker (which are both stored in an Excel-file) to a format that can be handled by the Main Module.
2) The *Main Module*, needed to calculate the distribution of attention over time and space, according to the three steps described in Section 4.2.
3) The *Visualization Module*, needed to plot the output of the Main Module in a format as shown in Figure 2.
4) The *Matlab to TTL Module*, needed to convert the output of the Main Module to traces that can be used to check properties on by the *TTL Checking Tool* and can be visualized by *LEADSTO* in the format shown in Figure 3.

In order to replicate the simulation results and automated verifications one has to do the following:

- Consider the Excel-file "data.xls" (see below) that contains raw experimental data.
- Import the data from the "targets", "gaze", and "actions" tabs in the Excel-file into *Matlab* and rename them to "inp1", "inp2", and "inp3", respectively.
- Load *Pre-Processing Module* in *Matlab* and run it. Wait for a while until all *Matlab* data is generated.
- Load *Main Module* in *Matlab* and run it. Wait for a while until all *Matlab* data is generated.
- Load *Visualization Module* in *Matlab* and run it. Wait until all plots have been processed and the file "movie_new.mpg" has been generated.
- Load *Matlab to TTL Module* and run it. Wait for a while until "attention.tr" has been generated.
- Install the *LEADSTO Simulation Tool* and the *TTL Checking Tool* from `http://www.cs.vu.nl/~wai/TTL/`.
- Open "attention.tr" in *LEADSTO* to view the trace. This takes a while.
- Use the *TTL Checking Tool* to check the properties in "ttl_check.fm" on "attention.tr".

The *Matlab* source code of the four modules of the model is provided in the subsections below. The data file "data.xls" and property file "ttl_check.fm" are downloadable from one of the authors' website[5].

[5]See `http://www.few.vu.nl/~pp/attention`.

Pre-Processing Module

```
% Pre-Processing Module
% By Tibor Bosse, Peter-Paul van Maanen, and Jan Treur

% ADAPT EXCEL SHEET

for i = 1:830 % depends on the time of the experiment
  for j = 1:4200 % depends on the length of the excel sheet
    if inp1(j,1) < 100*i && inp1(j+1,1) > 100*1
      for n = 1:size(inp1,2)
        input2(i,n) = inp1(j,n);
      end
    end
  end
end

for i = 1:830 % depends on the time of the experiment
  for j = 1:4200 % depends on the length of the excel sheet
    if inp2(j,3) < 100*i && inp2(j+1,3) > 100*1
      input1(i,1) = inp2(j,1);
      input1(i,2) = inp2(j,2);
    end
  end
end

% CREATE GAZE DATA FROM ADAPTED EXCEL SHEET

for i = 1:500 % or: size(input1,1)
  gaze(i,1) = max(0, ceil((input1(i,1)-220)/25));
  gaze(i,2) = max(0, ceil((input1(i,2)-55)/25));
end

% CREATE ACTION DATA FROM EXCEL SHEET

for i = 1:500 % or: size(input1,1)
  for j = 1:size(inp3,1)
    if i == ceil(inp3(j,3)/100)
      actions(i,1) = max(0, ceil((inp3(j,1)-220)/25));
      actions(i,2) = max(0, ceil((inp3(j,2)-55)/25));
    end
  end
end

% CREATE TRACK LOCATION DATA FROM ADAPTED EXCEL SHEET

mx = 31;
my = 28;

for i = 1:500 % or: size(input2,1)
  for x = 1:mx
    for y = 1:my
      if ceil((input2(i,2)-3240)/380) == x && ceil(input2(i,3)/380) == y
        tracks(x,y,i) = 1;
      elseif ceil((input2(i,10)-3240)/380) == x && ceil((input2(i,11)-240)/380) == y
        tracks(x,y,i) = 2;
      elseif ceil((input2(i,18)-3240)/380) == x && ceil((input2(i,19)-240)/380) == y
        tracks(x,y,i) = 3;
      elseif ceil((input2(i,26)-3240)/380) == x && ceil((input2(i,27)-240)/380) == y
        tracks(x,y,i) = 4;
      elseif ceil((input2(i,34)-3240)/380) == x && ceil((input2(i,35)-240)/380) == y
        tracks(x,y,i) = 5;
      elseif ceil((input2(i,42)-3240)/380) == x && ceil((input2(i,43)-240)/380) == y
        tracks(x,y,i) = 6;
      elseif ceil((input2(i,50)-3240)/380) == x && ceil((input2(i,51)-240)/380) == y
        tracks(x,y,i) = 7;
      elseif ceil((input2(i,58)-3240)/380) == x && ceil((input2(i,59)-240)/380) == y
        tracks(x,y,i) = 8;
      elseif ceil((input2(i,66)-3240)/380) == x && ceil((input2(i,67)-240)/380) == y
        tracks(x,y,i) = 9;
      else
        tracks(x,y,i) = 0;
      end
    end
  end
end

% CREATE TRACK ATTRIBUTE DATA FROM ADAPTED EXCEL SHEET

for i = 1:500 % or: size(input2,1)
  for t = 1:9
    % COLOUR
    colour(t,1,i) = t;
    if input2(i,4+8*(t-1)) == 1
      colour(t,2,i) = 9;
    elseif input2(i,4+8*(t-1)) == 2
      colour(t,2,i) = 5;
    elseif input2(i,4+8*(t-1)) == 3
      colour(t,2,i) = 8;
    elseif input2(i,4+8*(t-1)) == 4
      colour(t,2,i) = 4;
    elseif input2(i,4+8*(t-1)) == 5
      colour(t,2,i) = 3;
    elseif input2(i,4+8*(t-1)) == 6
      colour(t,2,i) = 3;
    else
```

```
95        colour(t,2,i) = 1;
      end
97    % DISTANCE
      distance(t,1,i) = t;
99    if input2(i,5+8*(t-1)) < 0
        distance(t,2,i) = 1;
101   else
        distance(t,2,i) = 10 - ceil(input2(i
            ,5+8*(t-1))/550);
103   end
      % TYPE
105   type(t,1,i) = t;
      if input2(i,6+8*(t-1)) == 1
107     type(t,2,i) = 6;
      elseif input2(i,6+8*(t-1)) == 2
109     type(t,2,i) = 8;
      elseif input2(i,6+8*(t-1)) == 3
111     type(t,2,i) = 4;
      else
113     type(t,2,i) = 1;
      end
115   % SPEED
      speed(t,1,i) = t;
117   if input2(i,7+8*(t-1)) < 0
        speed(t,2,i) = 1;
119   else
        speed(t,2,i) = ceil(input2(i,7+8*(t-1)
            )/100);
121   end
    end
123 end
```

Main Module

```
1  % Main Module
   % By Tibor Bosse, Peter-Paul van Maanen, and
        Jan Treur
3
   % CONSTANTS
5
   w1 = 0.8; % COLOUR
7  w2 = 0.5; % DISTANCE
   w3 = 0.1; % TYPE
9  w4 = 0.5; % SPEED
   a = 100;
11 d = 0.8;
   alph = 0.3;
13
   % INPUT-SPECIFIC CONSTANTS
15
   max_x = size(tracks, 1);
17 max_y = size(tracks, 2);
   end_time = size(actions, 1);
19 no_tracks = size(colour, 1);

21 % START SIMULATION

23 for i = 1:end_time

25 % CALCULATE CURRENT ATTENTION LEVEL
     for x = 1:max_x
```

```
27     for y = 1:max_y
         if tracks(x,y,i) == 0
29         v1 = 1;
           v2 = 1;
31         v3 = 1;
           v4 = 1;
33       else
           for x2 = 1:no_tracks
35           if tracks(x,y,i) == colour(x2,1,i)
               v1 = colour(x2,2,i);
37           end
           end
39         for x3 = 1:no_tracks
             if tracks(x,y,i) == distance(x3,1,
                 i)
41             v2 = distance(x3,2,i);
             end
43         end
           for x4 = 1:no_tracks
45           if tracks(x,y,i) == type(x4,1,i)
               v3 = type(x4,2,i);
47           end
           end
49         for x5 = 1:no_tracks
             if tracks(x,y,i) == speed(x5,1,i)
51             v4 = speed(x5,2,i);
             end
53         end
         end
55       current(x,y,i) = (v1*w1+v2*w2+v3*w3+v4
             *w4)/(1 + alph*((x - gaze(i,1))^2
             + (y - gaze(i,2))^2));
       end % note that I added a factor alph in
           the above formula
57   end

59 % CALCULATE TOTAL CURRENT ATTENTION LEVEL
     sum = 0;
61   for x = 1:max_x
       for y = 1:max_y
63       sum = sum + current(x,y,i);
       end
65   end

67 % NORMALISE ATTENTION LEVEL
     for x = 1:max_x
69     for y = 1:max_y
         normalised(x,y,i) = current(x,y,i)*a /
             sum ;
71     end
     end
73
   % CALCULATE REAL ATTENTION LEVEL
75   for x = 1:max_x
       for y = 1:max_y
77       if i == 1
           real(x,y,i) = normalised(x,y,i)*(1-d
               );
79       else
           real(x,y,i) = (real(x,y,i-1)*d) + (
               normalised(x,y,i)*(1-d));
81       end
       end
```

```
  end
end
```

Visualization Module

```
% Visualisation Module
% By Tibor Bosse, Peter-Paul van Maanen, and Jan Treur

% DISPLAY RESULTS

[X,Y] = meshgrid(1:max_y, 1:max_x);
for i = 1:end_time
  hold off;
  surf(X,Y,real(:,:,i));
  axis([0 28 0 31 0 8]);
%  campos([0 15 55]); % this can be used to modify the camera position
%  campos([17 14 55]); % from above
  campos([197 -22 49]); % from aside, viewed from the south
alpha(0.5);
  colorbar;
  hold on;
  for x = 1:max_x
    for y = 1:max_y
      for x6 = 1:no_tracks
        if tracks(x,y,i) == colour(x6,1,i) && colour(x6,2,i) == 9 % red
          plot3(y,x,0,'Color',[1 0 0],'Marker','.','MarkerSize',15);
        end
        if tracks(x,y,i) == colour(x6,1,i) && colour(x6,2,i) == 5 % flashing red
          plot3(y,x,0,'Color',[1 0 0],'Marker','.','MarkerSize',25);
        end
        if tracks(x,y,i) == colour(x6,1,i) && colour(x6,2,i) == 8 % yellow
          plot3(y,x,0,'Color',[1 1 0],'Marker','.','MarkerSize',15);
        end
        if tracks(x,y,i) == colour(x6,1,i) && colour(x6,2,i) == 4 % flashing yellow
          plot3(y,x,0,'Color',[1 1 0],'Marker','.','MarkerSize',25);
        end
        if tracks(x,y,i) == colour(x6,1,i) && colour(x6,2,i) == 3 % green
          plot3(y,x,0,'Color',[0 1 0],'Marker','.','MarkerSize',15);
        end
        if tracks(x,y,i) == colour(x6,1,i) && colour(x6,2,i) == 2 % (or 3) flashing green
          plot3(y,x,0,'Color',[0 1 0],'Marker','.','MarkerSize',25);
        end
      end
      if actions(i,1) == x && actions(i,2) == y
        plot3(y,x,0,'Color',[0 0 0],'Marker','.','MarkerSize',35);
      end
      if gaze(i,1) == x && gaze(i,2) == y
        plot3(y,x,0,'Color',[0 0 1],'Marker','*','MarkerSize',35);
      end
    end
  end
  pause(0.1);
end
```

Matlab to TTL Module

```
% Matlab to TTL Module
% By Tibor Bosse, Peter-Paul van Maanen, and Jan Treur
% July 2006

end_time = size(actions,1);

myfile = fopen('attention.tr', 'wt');
fprintf(myfile, 'times(0, %d, %d).\n', end_time+1, end_time+1);

% GAZE

for i = 1:end_time
  fprintf(myfile, 'atom_trace(_, gaze(%d, %d), [range(%d, %d, true)]).\n', gaze(i,1), gaze(i,2), i, i+1);
end

% ACTIONS

for i = 1:end_time
  if actions(i,1) > 0
    fprintf(myfile, 'atom_trace(_, mouse_click(%d, %d), [range(%d, %d, true)]).\n', actions(i,1), actions(i,2), i, i+1);
  end
end

% TRACKS

for i = 1:end_time
  for x = 1:size(tracks,1)
    for y = 1:size(tracks,2)
if tracks(x,y,i) > 0
          fprintf(myfile, 'atom_trace(_, is_at_location(%d, %d, %d), [range(%d, %d, true)]).\n', tracks(x,y,i), x, y, i, i+1);
        end
    end
  end
end
```

```
36
% ATTENTION
38
for i = 1:end_time
40   for x = 1:size(real,1)
       for y = 1:size(real,2)
42       fprintf(myfile, 'atom_trace(_,
             has_attention_level(%d, %d, %6.6f)
             , [range(%d, %d, true)]).\n', x, y
             , real(x,y,i), i, i+1);
       end
44   end
end
46
fclose(myfile);
```

Chapter 11

Design and Validation of HABTA: Human Attention-Based Task Allocator

This chapter appeared as (van Maanen et al., 2008a). Also an extended abstract (van Maanen et al., 2008b) appeared of this chapter.

Design and Validation of HABTA: Human Attention-Based Task Allocator

Peter-Paul van Maanen*†, Lisette de Koning* and Kees van Dongen*

* TNO Human Factors, P.O. Box 23, 3769 ZG Soesterberg, The Netherlands
Email: {peter-paul.vanmaanen, lisette.dekoning, kees.vandongen}@tno.nl
† Department of Artificial Intelligence, Vrije Universiteit Amsterdam
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands

***Abstract*—This paper addresses the development of an adaptive cooperative agent in a domain that suffers from human error in the allocation of attention. The design is discussed of a component of this adaptive agent, called Human Attention-Based Task Allocator (HABTA), capable of managing agent and human attention. The HABTA-component reallocates the human's and agent's focus of attention to tasks or objects based on an estimation of the current human allocation of attention and by comparison of this estimation with certain normative rules. The main contribution of the present paper is the description of the combined approach of design and validation for the development of such components. Two complementary experiments of validation of HABTA are described. The first experiment validates the model of human attention that is incorporated in HABTA, comparing estimations of the model with those of humans. The second experiment validates the HABTA-component itself, measuring its effect in terms of human-agent team performance, trust, and reliance. Finally, some intermediary results of the first experiment are shown, using human data in the domain of naval warfare.**

1 Introduction

Several challenges can be identified for work on future naval platforms. Information volumes for navigation, system monitoring, and tactical tasks will increase as the complexity of the internal and external environment also increases (Grootjen and Neerincx, 2005). The trend of reduced manning is expected to continue as a result of economic pressures and humans will be responsible for more tasks, tasks with increased load, and tasks with which they will have less experience. Problems with attention allocation are more likely to occur when more has to be done with less. To avoid these attention allocation problems, in this paper it is proposed that humans are supported by cooperative agents capable of managing their own and the human's allocation of attention. It is expected that these attention managers have a significant positive impact: when attentional switches between tasks or objects are often solicited, where the human's lack of experience with the environment makes it harder for them to select the appropriate attentional focus, or where an inappropriate selection of attentional focus may cause serious damage. In domains like air traffic control (ATC) or naval tactical picture compilation these properties are found, even when the people involved are experienced.

The present study discusses the design and validation of a component of an adaptive agent, called *Human Attention-Based Task Allocator (HABTA)*, capable of managing agent and human attention. This component is based on two cognitive models: one that describes the current allocation of a humans attention and one that prescribes the way his attention should be allocated. If there is a discrepancy between the output of the two models, HABTA reallocates the tasks between the human and the agent, for instance depending on certain rules the human and agent agreed upon. Models of attention or situation awareness have already been developed and used to predict faults in attention allocation (e.g., the SEEV model (Wickens et al., 2005)), but less is known about how they can be used to initiate agent adaptation, or automatic task reallocation more specifically. Furthermore, since in many domains (like ATC) it is the tasks altogether rather than mere visual stimuli that eventually require allocation of attention, the design and vali-

dation discussed in this paper is more focused on cognitive rather than visual attention. Off course the mentioned tasks also require visual attention, but all the time. Still other applied models mainly focus on visual attention. Finally, the applicability of a HABTA-based agent has not yet been investigated either.

This paper consists of the following sections. In Section 2 the psychological background of human error in the allocation of attention in the domain of naval warfare is shortly described. The understanding of these errors is important for the management of attention allocation. In Section 3 the design requirements of an agent-component *Human Attention-Based Task Allocator (HABTA)* are given. These requirements enable the agent to support the human-agent team by managing attention allocation of the human and the agent.

The main contribution of the present paper is the description of the combined approach of design and validation for the development of applied cooperative agent-components. In Section 4, two complementary methods of experimental validation against the in Section 3 stated design requirements are described. The first experiment validates the model of human attention that is incorporated in a HABTA-component. The validity of the model is determined by comparison of the model's and human's estimation of human attention allocation. The second experiment validates the HABTA-component itself, measuring its effect in terms of human-agent team performance, trust, and reliance. In Section 5 intermediary results of a pilot study are shown as a means to discuss the first experiment described in Section 4, using human data in the domain of naval warfare. In Section 6 the paper ends with concluding remarks and ideas for future research.

2 Human Error in the Allocation of Attention

As is mentioned in the introduction, the domain chosen in this research is naval warfare. One of the important tasks in naval warfare is the continuous compilation of a tactical picture of the situation (see for a description in more detail Chalmers et al., 2002). In a picture compilation task operators have to classify contacts that are represented on a radar display. The contacts can be classified as hostile, neutral or friendly, based on certain identification criteria (idcrits). Tactical picture compilation is known for its problems in the allocation of attention. To be able to identify contacts, contacts have to be monitored over time. This requires attention, but resources of attention are limited. When a task demands a lot of attention, less attentional resources are available for other tasks (e.g., Kahneman, 1973; Wickens, 1984). In general, two kinds of problems with human attention allocation can be distinguished: under-allocation of attention and over-allocation of attention.

Under-allocation of attention means that tasks or objects that need attention do not receive enough attention from the operator. *Over-allocation* of attention is the opposite: tasks or objects that do not need attention do receive attention. Over-allocation of attention to one set of tasks may result in under-allocation of attention to other tasks. Both under- and over-allocation of attention can lead to errors. Experience, training, and interface design can improve these limitations, but only to a certain level. Efforts have been done, for example, to fuse tactical information on displays (Steinberg, 1999). To be able to investigate whether a support system for attention allocation, like HABTA, can overcome these limitations of attention, it is important to understand these types of errors and more specifically in the domain of naval warfare. In Section 2.1 and 2.2, examples of errors of under- and over-allocation when performing a tactical picture compilation task and their possible causes are described.

2.1 Under-allocation of Attention

Under-allocation of attention means that some objects or tasks receive less attention than they need according to certain normative rules for the task to be performed. Under-allocation of attention occurs because of limited resources of attention or because of an incorrect assessment of the task.

When performing a tactical picture compilation task, operators have to monitor a radar screen where the surrounding contacts are represented as icons. The contacts on the screen have to be classified as neutral, hostile or friendly based on observed criteria. This is a complex task and it is essential that attention is allocated to the right objects. Inexperienced operators often allocate too little attention

to contacts that they have previously classified as neutral (Verkuijlen and Muller, 2007). When the behavior of these contacts changes to that of a hostile contact, this may not be observed because of under-allocation of attention to those contacts. One reason for this could be that identity changes are not expected by the operator due to the fact that people are too confident in their identified contacts. Another reason might be that changes in relevant behavior of contacts are not salient enough to be observed without paying direct attention to those objects. Under-allocation of attention to objects may also occur because of a lack of anticipatory thinking. This is the cognitive ability to prepare in time for problems and opportunities. In a picture compilation task, classification of contacts that are expected to come close to the own ship have priority over those that are not expected to come close. The reason for this is that there is less need to identify contacts when the own ship is out of sensor and weapon range of those contacts. Therefore, inexperienced operators often direct their attention only to objects in the direction the ship is currently heading. When unexpected course change is needed because of emerging threats, the ship is sometimes headed toward an area with contacts that are not yet classified (Verkuijlen and Muller, 2007).

2.2 Over-allocation of Attention

Apart from under-allocation, over-allocation of human attention is also a common problem. Over-allocation of attention means that some objects receive more attention than needed according to certain normative rules. Over-allocation of attention can occur for example, when operators overestimate the importance of a set of objects or tasks, while underestimating the importance of other objects or tasks. This occurs for example, when some contacts act like distractors and perform salient behavior. Comparable to visual search tasks where objects with salient features generate a pop-out effect (e.g., Treisman, 1993), those contacts directly attract the attention of the operator (bottom-up). Especially inexperienced operators overrate those salient cues and allocate too much attention to those contacts (Verkuijlen and Muller, 2007). Another possibility is that irrelevant behavior of objects is highly salient due to the manner information is presented on the interface. For instance, when a contact's behavior is unexpected, but not threatening, attention is unnecessarily drawn to this contact. In this case, the correct and quick application of identification rules will result in neutral identity and resources become available for the identification of other contacts.

3 Design Requirements

The goal of the efforts described is to come to a generic methodology for developing a component for an agent that supports humans with the appropriate allocation of attention in a domain that suffers from human error in the allocation of attention. As mentioned in Section 2, human attention allocation is prone to two types of errors with several possibilities as causes, such as inexperience and information overload.

In this section the design requirements of an agent-component is described that enables agents to determine whether objects or tasks that are required to receive attention indeed do receive attention, either by the human or the agent, and to intervene accordingly. The component is called an *Human Attention-Based Task Allocator (HABTA)*-component, since it *bases* its decisions to intervene on estimations of *human attention* and intervenes by *reallocating tasks* to either human or agent. It is expected that the combined task performance of the human-agent team will be optimized when the agent consists of such a HABTA-component. This work builds forth on earlier work. In (Bosse et al., 2007a) some of the possibilities are already discussed of dynamically triggering task allocation for tasks requiring visual attention, and in (Bosse et al., 2006, 2007b) the real-time estimation of human attentional processes in the domain of naval warfare is already discussed.

Properly stated design requirements are important for the design of effective agent-systems for a certain purpose and for validating whether the design meets the requirements for that purpose. A HABTA-component has four design requirements, which are the following:

1) It should have a descriptive model, meaning an accurate model of what objects or tasks in the task environment receive the human's attention,

2) It should have a prescriptive or normative model, meaning an accurate model of what objects require attention for optimal task performance,
3) It should be able to reliably determine whether actual attention allocation differs too much from the required attention allocation,
4) It should be able to support by redirecting attention or by taking over tasks such that task performance is improved.

In Figure 1 the design overview of a HABTA-component is shown that corresponds to the above design requirements. The setting in this particular overview is a naval officer behind an advanced future integrated command and control workstation and compiling a tactical picture of the situation. If the agent cooperatively assists the officer, than the agent should have a descriptive (Requirement 1) and normative model (Requirement 2). When the operator allocates his attention to certain objects or tasks that also require to receive attention, the outcome of both models should be comparable. This means that output of the models should not differ more than a certain threshold. The output of the two models in the example shown in Figure 1 are clearly different: in the left image, the operator is attending to different objects and corresponding tasks than the right image indicates as being required (see arrows). Because of this discrepancy, which the HABTA-component should be able to determine (Requirement 3), an adaptive reaction by the agent is triggered (Requirement 4). This means that, for instance, the agent either will draw attention to the proper region or task through the workstation, or it will allocate its own attention to this region and starts executing the tasks related to that region, for the given situation.

To prevent that HABTA-based support results in automation surprises, the human-agent team should be able to make and adjust agreements about how they work as a team. It may be, for example, that the human does not want to be disturbed, and the agent is supposed to allocate tasks solely to itself. This option requires a higher form of autonomous task execution by the agent. The other possibility is that the human wants to stay in control as much as possible and therefore only wants to be alerted by the agent to attend to a certain region or execute a certain task. The choice of the agent's autonomy or assertiveness can also depend on a certain estimate of the urgency for reallocating tasks. In the case of tactical picture compilation, human and agent should agree on whether the agent is allowed to take over identification tasks for contacts that are overlooked or not.

On the one hand, the human may be preferred to be dealing with an arbitrary region or task, because the human may have certain relevant background knowledge the agent does not have. But on the other hand, the human is not preferred to be allocated to all objects or tasks at once, because, in a complex scenario, he has limited attentional resources. Hence humans cannot be in complete control, given the fact that both human and agent need each other. Optimal performance is only reached when human and agent work together as a team. Human-agent team work is expected to be effective when the right support is provided at the right time and in the right way. An obvious goal, but there are some potential obstacles in achieving it. Descriptive and prescriptive (normative) models of attention allocation may be inaccurate. Objects that require or receive attention may not be in the output of the descriptive or normative models, respectively. Similarly, objects that do not require or receive attention may be in the output of the models. The agent may conclude that descriptive and normative models differ when they do not, and vice versa. The system may be assertive and wrong, or withholding but right. Attention may be redirected to the wrong region or the wrong set of objects, or tasks are taken over by the agent that should be taken over by the human. Because of the complexity of these consequences of the above design requirements, both the validity of the model and the effectiveness of the agent's HABTA-component should be investigated and iteratively improved. This procedure of investigation and improvement is described in Section 4.

4 Validation

As described in Section 3, HABTA-components require a descriptive and prescriptive model of attention to support attention allocation of humans in complex tasks. Before HABTA-components can be used to support humans, they have to be validated. Validation is the process of determining the degree

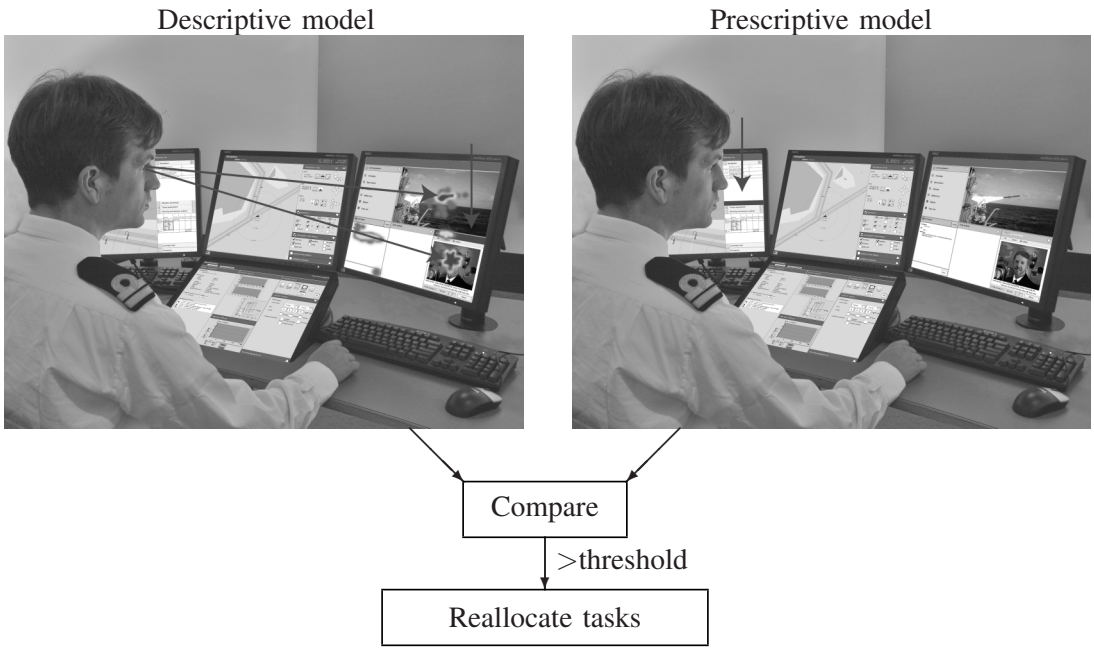


Figure 1. Design overview of a HABTA-component for a future integrated command and control environment. The discrepancies between the output of the descriptive and prescriptive model result in a reallocation of tasks. The workstation shown in the pictures is the Basic-T (Arciszewski and van Delft, 2005).

to which a model is an accurate description of certain real world phenomena from the perspective of the intended use of the model. Again referring to Section 3, for the intended use mentioned in this paper, this means that HABTA-components have to meet the design requirements (1–4) in Section 3.

In the near future two experiments will be carried out to validate a HABTA-component. In Experiment 1 the descriptive model will be validated (Requirement 1). This experiment aims at determining the sensitivity (d') of the model by comparing it with data retrieved from human subjects executing a complex task that causes problems with attention allocation. Based on the results of the experiment, the d' of the model can be improved by optimizing it off-line against a random part of the same data. It is expected that higher d' results in better support based on the HABTA-component. If the d' of the descriptive model is not high enough, the HABTA-component will consequently support at the wrong moments and for the wrong reasons, which obviously leads to lower performance, trust, and acceptance. In Experiment 2 the applicability of the (improved) descriptive model for attention allocation support is tested (Requirements 2–4). It will be investigated if the support of an agent with the HABTA-component leads to better performance than without HABTA-component.

The remainder of this section is composed of three parts. In Section 4.1 the task that will be used in the above mentioned experiments is described in more detail. After that, the specific experimental design and measurements of the experiments are described in Sections 4.2 and 4.3, respectively. Both experiments still have to be carried out. Preliminary results from a pilot of Experiment 1 will be described in Section 5.

4.1 Task Description

The task used in both experiments is a simple version of the identification task described in (Heuvelink and Both, 2007) that has to be executed

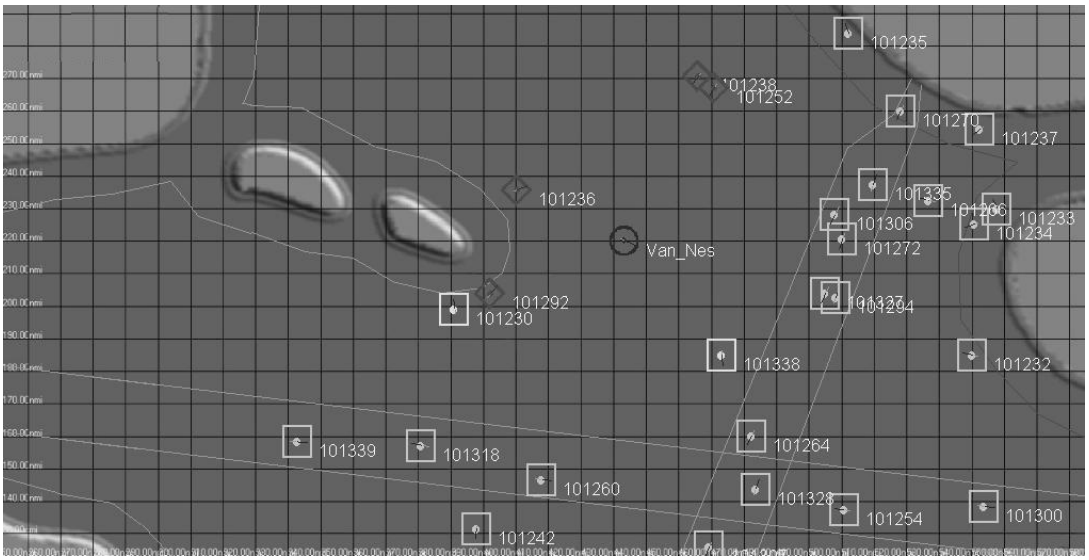


Figure 2. The interface of the used simplified task environment based on (Heuvelink and Both, 2007). The long lanes are sea lanes. The circle labeled with "Van_Nes" represents the own ship.

in order to buildup a tactical picture of the situation. In Figure 2 a snapshot of the interface of the task environment is shown. The goal is to identify the five most threatening contacts (ships). In order to do this, participants have to monitor a radar display where contacts in the surrounding areas are displayed. To determine if a contact is a possible threat, different criteria have to be used. These criteria are the identification criteria (idcrits) that are also used in naval warfare, but are simplified in order to let naive participants learn them more easily. These simplified criteria are the speed, heading, distance of a contact to the own ship, and whether the contact is in a sea lane or not. If a participant concludes that a ship is a possible threat or not, he can change the color of the contacts by clicking with the left mouse button on the contact. Contacts can be identified as either a threat (red), possible threat (yellow), or no threat (green). It is not necessary that all contacts are identified. Only the five most threatening have to be identified as a threat (marked as red). The other types of identification (possible threat and no threat) are used to assist the participant in his task. When a contact is marked as green, this means no direct attention is needed. When a contact is marked as yellow, this contact has to be checked regularly to decide if the contact is still no threat. The task has to be performed as accurately as possible. Contacts that are wrongfully identified as a threat will result in a lower score. Performance is determined by the accurateness, averaged over time, of the contacts that are identified as the five most threatening contacts during the task. Behavior of each contact can change during the task and therefore the soundness of classifications (which is not communicated to the participant) may change over time. For instance, a contact can suddenly come closer to the own ship, get out of a sea lane, speedup, or change its heading.

For Experiment 2 (see Section 4.3) the task is extended to one that includes the support of the HABTA-based agent. The support agent is capable of doing the same as the operator, except with limited background knowledge and therefore limited performance per object. In order to simulate this aspect, for each participant, the measured average performance per contact in Experiment 1 is used in order to set the performance of the agent. The agent can be given a list of objects provided by the its HABTA-component and compile a tactical picture

related to those objects.

Apart from the primary task mentioned so far, in both experiments a secondary task is used in order to control for the attentional resources. This secondary task is a gauge task that has to be monitored constantly. The secondary task is shown on a different screen and requires action on various occasions depending on the value the gauge is indicating. The primary and secondary task performances are equally important in order to have the gauge task be effective.

4.2 Experiment 1: Validation of the Descriptive Model

In Experiment 1 participants perform the task as described in the previous section without support of the agent. The same scenarios will be used for all participants. Before the actual task starts, the task will be explained thoroughly to the participants. The task will be illustrated by using different examples to be sure that the participants understand the task and how to decide if a contact is a threat based on the different criteria. All participants have to perform a test to check if they sufficiently understood the rules of classifying the contacts. If they do not perform well, i.e., their score is below 80%, they receive extra instructions and another test. Also the possible second test has to be performed with a success rate of above or equal to 80%. Then they have to perform a practice trial in which they have to apply the learned rules. After this they get instructions of how to behave during the experimental interventions while they are executing their task. This is practiced as well for several times, after which the actual experiment begins. This introduction is necessary because we do not want the participants to perform badly because of a lack of understanding of the rules or experience with the experimental interface.

During the task, different variables are measured to determine the d' of the model and to be able to iteratively improve the model afterwards. The following variables will be measured: eye movements, performance of the primary and secondary task, mental workload, and at different points in time participants have to mark contacts that received attention according to the themselves. The performances and mental workload measures are used as a baseline for comparing the performances and mental workload of the task with and without using the HABTA-component (see Experiment 2). In order to measure the variables, at random moments (varying from 2–6 minutes) the scenario is frozen. There are two experimental sessions, one with an easy and one with a difficult scenario of half an hour. It is expected that the difficult scenario would also lead to a poorer performance of the model, because in those cases the participant's allocation of attention is less dynamic. During a freeze, the participants have to click on the contacts to which, in their opinion, they had allocated their attention to from 3 seconds before the scenario was frozen. The participants also have to motivate why those contacts are selected. Directly after the participant has selected the contacts, mental workload is measured during the same freezes. For this, the mental workload scale from (Zijlstra and van Doorn, 1985) is used (BSMI). On a scale from 0 (not at all strenuous) to 150 (very strenuous) the mental workload of the task has to be indicated. Performance and eye movements of the participants are measured during the task, by calculation according to the rules described in Section 4.1 and by eye-tracker recording, respectively. The patterns of the eye movements (what objects are looked at through time) are compared with the contacts that received attention before the freezes, according to the participant. This is done to be sure that the participants were able to select the objects that received their attention. Those contacts that got a considerate amount of gaze fixations, are expected to have received attention.[1] If the participants do not mention those contacts, it is expected that they are not good at selecting the proper contacts.

After the experiment is performed, the contacts selected by the participants during the freezes are matched with the output of the model in a simulation. The calculation of d' provides information about the sensitivity of the model, i.e., whether the model is able to accurately describe the participant's dynamics of attention allocation. Information about performances, workload, and the description of the participants why contacts are selected, is expected to be valuable for determining in what cases the per-

[1]Note that this does not hold vice versa, which would otherwise mean that attention in complex scenarios is easily described using solely fixation data.

centage true positives (hits) is high and percentage false positives (false alarms) is low, which in turn can be used to improve the sensitivity of the model.

4.3 Experiment 2: Validation of the HABTA-Based Support

In Experiment 2 the applicability of the model for supporting attention allocation is tested. The same task as in Experiment 1 has to be performed, except this time the participant is supported by the agent of which the HABTA-component is part of. When there is a discrepancy between the descriptive and prescriptive model, higher than a certain threshold (see Figure 1), the agent will support the human by either performing the task for the participant or by drawing attention to the contact that should receive attention. Different variables are measured to determine the excess value of the HABTA-based support. Performances and mental workload are measured in the same way as in Experiment 1. Furthermore, trust and acceptance are measured at the end of the scenario. In order to determine the effectiveness of an agent, it is important to measure trust and acceptance of that agent and to investigate what factors influence trust and acceptance. Trust and acceptance indicate whether people will actually use the agent. For instance, it says something about whether people will follow the advice of the agent, in the case the agent provides advice. Validated questionnaires are adjusted (only when necessary) to be able to measure trust and acceptance in adaptive systems. The trust questionnaire is based on the questionnaire of (Madson and Gregor, 2000). An example of a question on this questionnaire is: "Is the agent reliable enough?". The acceptance questionnaire is based on the questionnaire of (Venkatesh et al., 2003) and (Davis, 1989). An example of a question on this questionnaire is: "Is the support of the agent useful for me?". The trust and acceptance scores are expected to provide more insight in the results of the experiment. If trust in and acceptance of the agent is low, people will not follow any suggestions made by the agent.

The performance and mental workload without a HABTA-based agent will be compared with those with a HABTA-based agent, using the results of Experiment 1 as a baseline. This is one of the reasons that the same participants are used as in Experiment 1. The other reason is that the measured performance in Experiment 1 is used for setting the performance of the agent. For Experiment 2, it is expected that performance is higher and mental workload is lower when supported with HABTA.

5 Intermediary Results

In this section preliminary results of the experiments described in Section 4 are shown based on a pilot study for Experiment 1, using one arbitrary participant. The actual experiment will be performed with more participants. The pilot is primarily meant to explore the applicability of the experimental method of Experiment 1 to the given task. It is also meant as an illustration of the form and dynamics of the participant's and model's estimation of human allocation of attention. Finally, it is used as a basis for a better understanding of the possibilities of HABTA-based support, which is important for a proper preparation and performing of Experiment 2. This is because this type of support is required in the experimental setup of Experiment 2.

In the pilot study, the participant was required to execute the identification task and to select contacts during the freezes. In contrast with the procedure during the actual experiment, no questions concerning the participant's cognitive workload or motivation for the selected contacts were asked. In Figure 2 the interface right before a freeze is shown. During a freeze both the participant and the model had to indicate their estimation of what contacts the attention of the participant was allocated to. In the situation presented in Figure 2, the participant selected contacts 101238, 101252, 101236, 101338, 101230, 101292, 101294, and 101327. Between every two freezes certain events can cause the participant to change the allocation of his attention to other attention demanding regions. The preceding course of events of the situation in Figure 2 clearly caused the participant to attend to the contacts close to his own ship "Van_Nes". If the model made a proper estimation of the participant's allocation of attention, the selected contacts by the participant would resemble those selected by the model. Consequently, the performance of the model is best determined by means of the calculation of the overall overlap of the participant's and model's selection of contacts. This calculation is explained below.

Table I
CONFUSION MATRIX OF THE PARTICIPANT'S AND MODEL'S ESTIMATION OF THE ALLOCATION OF ATTENTION.

		Participant		
		t	f	*total*
Model	t'	Hits	False Alarms	T'
	f'	Misses	Correct Rejections	F'
	total	T	F	

There are four possible outcomes when comparing the participant's and model's selection of contacts, namely, a Hit, False Alarm, Correct Rejection, and Miss. The counts of these outcomes can be set out in a 2×2 confusion matrix. Table I is such a confusion matrix, where T and F are the total amount of the participant's selected and not selected contacts, respectively, and T' and F' are the total amount of the model's selected and not selected contacts, respectively. The ratios of all the possible outcomes are represented by H, FA, CR, and M, respectively. A higher H and CR, and a lower FA and M, leads to a more appropriate estimation by the model. This is the case because the selected contacts by the model then have a higher resemblance with those selected by the participant. Furthermore, a higher T' leads to a higher H, but, unfortunately, also to a higher FA. Something similar holds for F'. The value of T' therefore should depend on the trade-off between the costs and benefits of these different outcomes.

In Figure 3 the output of the model for the situation presented in Figure 2 is shown. If the estimated attention on the z-axis, called Attention Value (AV), is higher than a certain threshold, which is in this case set to .035, the contact is selected and otherwise it is not. The different values of AV are normally distributed over the (x, y)-plane. The threshold is dependent on the total amount of contacts the participant is expected to allocate attention to (Bosse et al., 2006). The AV-distribution in Figure 3 results in the selection of contacts 101235, 101238, 101252, 101236, 101292, 101230, 101338, and 101260. Using this selection and the selected contacts by the participant, for each contact, the particular outcome can be determined. For each freeze, if one counts the number of the different outcomes, a confusion matrix can be constructed and the respective ratios can be calculated. For Figure 3, for example, these ratios are $H = \frac{6}{8} = 0.750$, $FA = \frac{2}{19} = 0.105$, $CR = \frac{17}{19} = 0.895$, and $M = \frac{2}{8} = 0.250$, respectively.

To study the performance of models Receiver-Operating Characteristics (ROC) graphs are commonly used. A ROC-space is defined by FA as the x- and H as the y-axis, which depicts relative trade-offs between the costs and benefits of the model. Every (FA, H)-pair of each confusion matrix represents one point in the ROC-space. Since the model is intended to estimate the participant's allocation of attention for each freeze and participant, this means that for N participants and M freezes, there are NM points in the ROC-space.

Once all points have been scatter plotted in the ROC-space, a fit of an isosensitivity curve leads to an estimate of the d' of the model. Isosensitivity corresponds to:

$$d' = z(H) - z(FA)$$

where d' is constant along the curve and $z(x)$ is the z-score of x.[2] Larger absolute values of d' mean that the model is more specific and sensitive to the participant's estimation (and thus has a higher performance). If d' is near or below zero, this indicates the model's performance is equal to or below chance, respectively. If there does not exist a proper fit of a isosensitivity curve, the area under the curve (AUC) can also be used as a model validity estimate. In non-parametric statistics the ROC-graph is determined by the data and not by a predefined curve. If the different values of H and FA appear to be normally distributed, the d' can be obtained from a z-table. In this case, the (FA, H)-pair from Figure 3 results in $d' = 1.927$. Which is a fairly good score.

6 CONCLUSION AND DISCUSSION

This paper describes the development of an adaptive cooperative agent to support humans while performing tasks where errors in the allocation of

[2]The z-score reveals how many units of the standard deviation a case is above or below the mean:

$$z(x_i) = \frac{x_i - \mu_x}{\sigma_x}$$

where μ_x is the mean, σ_x the standard deviation of the variable x, and x_i a raw score.

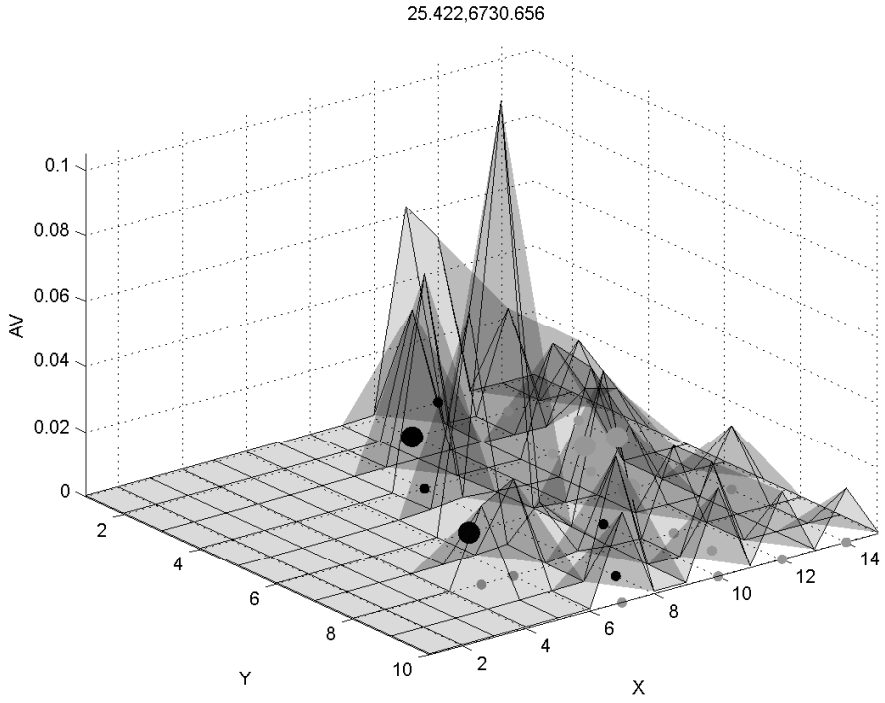


Figure 3. The output of the model for the situation shown in Figure 2. The black dots are the selected contacts by the model. Bigger dots mean that there are more contacts on the respective coordinates.

attention occur. In general, human attention allocation is prone to two types of errors: over- and under-allocation of attention. Several factors may cause over- or under-allocation of attention, such as inexperience and information overload. The design is discussed of a component of an agent, called Human Attention-Based Task Allocator (HABTA), that is capable of detecting human error in the allocation of attention and acts accordingly by reallocating tasks between the human and the agent. In this way the HABTA-based agent manages human and agent attention, causing the performance of the human-agent team to increase. The development of such an agent requires extensive and iterative research. The agent's internal structure, i.e., the models describing and prescribing human attention allocation and the support mechanism that is based on those models, has to be validated. In this paper, two experimental designs are described to validate the internal of the agent. The first experiment aims at validating the model of human attention allocation (descriptive model) and the second experiment aims at validating the HABTA-component as a whole, incorporating a prescriptive model and support mechanism.

The results from the pilot of the first experiment presented in this paper have proven to be useful, but the actual experiments still have to be performed. Therefore, future research will focus on the performance and analysis of these experiments. It is expected that the accuracy of the model can be increased hereafter, however 100% accurateness will not be attainable. The results of the first experiment will show if the variables indeed provide enough information to improve the accurateness of the model.

With respect to the second experiment, one might argue to add another variant of support, such as

one that is configured by the participant itself. The participant will then do the same as HABTA does, which might result in him being a fair competitor for HABTA. In this way the effectiveness of HABTA-based support can be studied more convincingly, comparing human-agent performance when either the participant or the agent is managing attention allocation. Deciding on this will be subject in the near future.

If the agent does not support the human at the right time and in the right way, this might influence trust and acceptance of the agent. It is interesting to investigate whether an observable and adjustable internal structure of the agent improves trust and acceptance of the system (e.g., Mioch et al., 2007) in these proceedings). This also needs further research.

In this paper the development and validation of a normative model (prescriptive model) is not described. Validation of this model is important, as it is also a crucial part of the HABTA-component. Errors in this model will lead to support at the wrong time and this will influence performance, trust, and acceptance. Further research is needed in order to develop and validate normative models.

Finally, in general, agent-components have more value when they can be easily adjusted for other applications. It is therefore interesting to see whether HABTA-based support can be applied in other domains as well.

Acknowledgments

This research was funded by the Dutch Ministry of Defense under progr. nr. V524. The authors would like to thank Annerieke Heuvelink, Harmen Lafeber, Jan Treur, Kim Kranenborg, Marc Grootjen, Tibor Bosse, Tjerk de Greef, and Willem van Doesburg for their contribution and helpful comments.

References

Arciszewski, H. F. R. and van Delft, J. H. (2005). Automated crew support in the command centre of a naval vessel. In *Proceedings of the 10th International Command and Control Research and Technology Symposium*.

Bosse, T., van Doesburg, W., van Maanen, P.-P., and Treur, J. (2007a). Augmented metacognition addressing dynamic allocation of tasks requiring visual attention. In Schmorrow, D. D. and Reeves, L. M., editors, *Proceedings of the Third International Conference on Augmented Cognition (ACI) and 12th International Conference on Human-Computer Interaction (HCI'07)*, volume 4565 of *Lecture Notes in Computer Science*. Springer Verlag.

Bosse, T., van Maanen, P.-P., and Treur, J. (2006). A cognitive model for visual attention and its application. In Nishida, T., editor, *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT-06)*, pages 255–262. IEEE Computer Society Press.

Bosse, T., van Maanen, P.-P., and Treur, J. (2007b). Temporal differentiation of attentional processes. In Vosniadou, S. and Kayser, D., editors, *Proceedings of the Second European Cognitive Science Conference (EuroCogSci'07)*, pages 842–847. IEEE Computer Society Press.

Chalmers, B. A., Webb, R. D. G., and Keeble, R. (2002). Modeling shipboard tactical picture compilation. In *Proceedings of the Fifth International Conference on Information Fusion*, volume 2, pages 1292–1299, Sunnyvale, CA. International Society of Information Fusion.

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3):319–340.

Grootjen, M. and Neerincx, M. (2005). Operator load management during task execution in process control. In *Human Factors Impact on Ship Design*.

Heuvelink, A. and Both, F. (2007). Boa: A cognitive tactical picture compilation agent. In *Proceedings of the 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2007)*, page forthcoming. IEEE Computer Society Press.

Kahneman, D. (1973). *Attention and effort*. Prentice Hall, Englewoods Cliffs, NJ.

Madson, M. and Gregor, S. (2000). Measuring human-computer trust. In *Proceedings of the Australasian Conference on Information Systems*.

Mioch, T., Harbers, M., van Doesburg, W. A., and van den Bosch, K. (2007). Enhancing human understanding through intelligent explanations. In Bosse, T., Castelfranchi, C., Neerincx, M., Sadri, F., and Treur, J., editors, *Proceedings of the*

first international workshop on human aspects in ambient intelligence.

Steinberg, A. (1999). Standardisation in data fusion. In *Proceedings of Eurofusion'99: International Conference on Data Fusion*, pages 269–277, UK. Stratford-Upon-Avon.

Treisman, A. (1993). The perception of features and objects. *Attention: Awareness, selection, and control*.

Venkatesh, V., Morris, M. G., Davis, G. B., and Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. *MIS Quarterly*, 27(3):425–478.

Verkuijlen, R. P. M. and Muller, T. J. (2007). Action speed tactical trainer review. Technical report, TNO Human Factors.

Wickens, C., McCarley, J., Alexander, A., Thomas, L., Ambinder, M., and Zheng, S. (2005). Attention-situation awareness (a-sa) model of pilot error. Technical Report AHFD-04-15/NASA-04-5, University of Illinois Human Factors Division.

Wickens, C. D. (1984). Processing resources in attention. In Parasuraman, R. and Davies, D. R., editors, *Variaties of attention*, pages 63–101, Orland, FL. Academic Press.

Zijlstra, F. R. H. and van Doorn, L. (1985). *The Construction of a Scale to Measure Perceived Effort*. Department of Philosophy and Social Sciences, Delft University of Technology, Delft, The Netherlands.

Chapter 12

Effects of Task Performance and Task Complexity on the Validity of Computational Models of Attention

This chapter appeared as (de Koning et al., 2008).

Effects of Task Performance and Task Complexity on the Validity of Computational Models of Attention

Lisette de Koning*, Peter-Paul van Maanen*† and Kees van Dongen*

* TNO Human Factors, P.O. Box 23, 3769 ZG Soesterberg, The Netherlands
Email: {lisette.dekoning, peter-paul.vanmaanen, kees.vandongen}@tno.nl
† Department of Artificial Intelligence, Vrije Universiteit Amsterdam
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands

***Abstract*—Computational models of attention can be used as a component of decision support systems. For accurate support, a computational model of attention has to be valid and robust. The effects of task performance and task complexity on the validity of three different computational models of attention were investigated in an experiment. The gaze-based model uses gaze behavior to determine where the subject's attention is, the task-based model uses information about the task and the combined model uses both gaze behavior and task information. While performing a tactical compilation task, participants had to indicate to what set of objects their attention was allocated. The indications of the participants were compared with the estimations of the three models. The results show that overall, the estimation of the combined model was better than that of the other two models. Contrary to what was expected, the performance of the models was not different for good and bad performers and was not different for a simple and complex scenario. The difference in complexity and performance might not have been strong enough. Further research is needed to determine if improvement of the combined model is possible with additional features and if computational models of attention can effectively be used in decision support systems. This can be done using a similar validation methodology as presented in this paper.**

1 Introduction

In the domain of naval warfare, information volumes for navigation, system monitoring and tactical tasks will increase while the complexity of the internal and external environment also increases (Grootjen and Neerincx, 2005). The trend of reduced manning is expected to continue as a result of economic pressures and humans will be responsible for more tasks with increased workload. Attention can be divided between different tasks, however attentional resources are limited (Wickens, 1984; Kahneman, 1973). Experience, training and interface design can improve these limitations, but only to a certain level. Even with experienced users, attentional problems are still likely to occur (Pavel et al., 2003). In naval warfare, errors caused by attentional problems can have serious consequences. Automation can assist the human by directing attention to critical events via alarms and alerts (Wickens and McCarley, 2007) or by taking over tasks. Knowing when a user needs support, a cognitive model of attention can be used. A cognitive model of attention is a model that estimates where the attention of the user is allocated at a certain moment. Together with a normative model that estimates where attention should be allocated, a decision support system can aid the user in dividing limited attentional resources. When these models are not accurate, support occurs at the wrong place and the wrong time. This will affect trust and acceptance of the decision support system and subsequently reliance behavior (Parasuraman and Riley, 1997; Dzindolet et al., 2003). To be able to develop a valid cognitive model of attention it is important to know what information can be used to estimate where attention is allocated and what factors affect the validity of a cognitive model of attention.

In the following section we will discuss what information is useful in estimating where attention is allocated. This will lead to three different cognitive

models of attention. After that, we will discuss the effects of task complexity and performance on the validity of the three models and describe the corresponding hypotheses. The focus in this paper will be on the tactical picture compilation task that is performed in naval warfare. The goal of a tactical picture compilation task is to build up a situation of surrounding ships (contacts).

1.1 Allocation of Attention

The direction of eye gaze is informative about where attention is directed (Just and Carpenter, 1976; Salvucci, 2000). In search tasks, eye movements may be indicative of where attention is allocated. A tactical picture compilation task is comparable with a search task. Targets (hostiles) have to be identified between different distractors (for example, container ships). In directing attention, a distinction has to be made between overtly orienting attention and covertly orienting attention. Overt changes in directing attention can be observed by head movements and eye movements. Covert orienting means directing attention to a location other than the one where the eyes are fixated and cannot be measured by eye and head movements (Posner, 1980). To be able to estimate where attention is allocated, gaze behavior will not be sufficient. Besides gaze behavior, knowledge about how a task is expected to be performed may also be indicative of where attention is allocated. The goal of a task will affect where attention will be allocated in a top-down manner (Treisman and Galade, 1980) and will affect both covert and overt allocation of attention. When the goal of the task is to keep track of green objects, attention may be directed overtly to those objects, but when the targets are found it is possible to track those targets covertly. Information about the goal of the task will provided additional information about where attention is allocated next to eye-movements. Besides eye-movements and information about the task, saliency of different stimuli of the task (e.g., color or brightness) may also be informative about where attention is allocated. Salient objects may attract attention in a bottom-up manner (Ouerhani et al., 2004). However, it is expected that this effect will be minimal in a tactical picture compilation task. Features that might capture attention in a bottom-up fashion, for example high speed of a contact, are also expected to receive attention based on the goal of the task. Contacts with a high speed are more threatening than contacts with a lower speed and will therefore receive attention. To determine how informative gaze behavior, characteristics of the task and the combination of both are to estimate where attention is allocated, three different cognitive models of attention were developed. The first model, the gaze-based model, uses gaze behavior to estimate where attention is allocated. The second model, the task-based model, uses information about how the task is expected to be performed. The third model, the combined model, uses both types of information to estimate where attention is allocated. Task complexity and task performance affect allocation of attention and are also expected to affect the validity of the three cognitive models. In the following sections, the possible effects of these variables on the validity of the cognitive models will be discussed.

1.2 Task Complexity

Task complexity is related to multiple features of a task, for example having to deal with rapidly evolving situations, cognitive complexity and uncertain data (Wood, 1986). Considering the characteristics of a picture compilation task, the complexity of a task can be determined by the *dynamics, ambiguity* and *volume* of information. Information is *dynamic* when the type, semantics, or volume of information varies over time. Information is *ambiguous* when the information which is needed to perform the task is unclear, incomplete, contradictive or inaccurate. The *volume* of information refers to the amount of information or events that occur at the same time. When task complexity increases, for example because of ambiguous information or more contacts, more attentional resources are needed to identify the contacts and it will be harder to allocate attention to the right contacts. Estimating where attention of the user is allocated is also more difficult. This leads to the following hypothesis:

Hypothesis 1. *The validity of all three models is higher in a simple than in a complex task.*

The combined model uses more information to estimate where attention is allocated than the other two models, namely information about gaze behav-

ior and information about how the task is expected to be performed. Using only one type of information will not be sufficient to estimate where attention is allocated. This leads to the following hypothesis:

Hypothesis 2. *For both complex and simple tasks, the validity of the combined model is higher than both the task- and the gaze-based models.*

Use of more information is of extra value in complex tasks, because in complex tasks it is harder to estimate where attention is allocated. This results in the following hypothesis:

Hypothesis 3. *The difference in validity between the combined model and the task- and gaze-based model is higher in a complex than in a simple task.*

1.3 Task Performance

How well people perform a certain task affects the allocation of their attention. People that are more experienced will be better at dividing attention between different sources of information. Research on the effects of playing video games has shown that our visual attention abilities may improve with training. Experienced players of video games required less attentional resources for a given target (Green and Bavelier, 2003). When performing a tactical picture compilation task, experts will be able to track more contacts. Experts will also be able to determine more quickly whether a contact is a possible threat. As opposed to poor performers, good performers will apply the rules correctly. The allocation of attention of good performers will be very similar to the task-based model. For poor performers, the allocation of attention will differ from the estimate of the task-based model. The combined model is partly based on the task-based model. This results in the following hypothesis:

Hypothesis 4. *The validity of the combined and the task-based model is higher for good performers than for poor performers.*

The combined model uses more information to estimate where attention is allocated than the other two models, namely information about gaze behavior and information about how the task should be performed. This leads to the following hypothesis:

Hypothesis 5. *For both good and poor performers, the validity of the combined model is higher than both the task- and the gaze-based models.*

When hypothesis 2 is true and when hypothesis 5 is true, this leads to the following hypothesis:

Hypothesis 6. *The validity of the combined model is higher than both the task- and the gaze-based models.*

2 Method

2.1 Participants

42 College students (22 male, 20 female) with an average age of 23 years ($SD = 2.29$) participated in the experiment as paid volunteers.

2.2 Task

Participants had to perform a dual task. The first task is derived from the tactical picture compilation task that is performed in naval warfare. The goal of the tactical picture compilation task is to build up awareness of threats surrounding the ships (contacts). In Figure 1 a screen-shot of the interface of the task environment is shown.

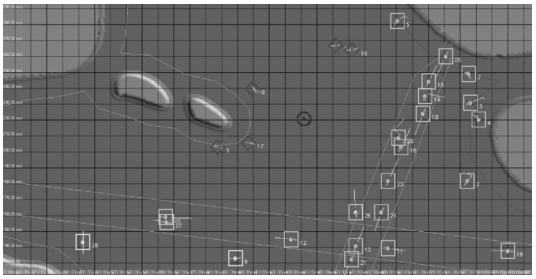


Figure 1. Radar screen with different contacts.

Participants had to identify the five most threatening contacts and mark them red by clicking on them. To determine if a contact is a possible threat the following criteria had to be used: speed, heading, distance of a contact to the own ship and whether a contact was positioned in a sea-lane or not. The behavior of the contacts was such that they varied on these criteria, which made them more or less threatening over time. For instance, a contact could get out of a sea lane, speedup, or change its heading

toward the own ship. Contacts that were mistakenly identified as a threat (false alarm) or contacts that were mistakenly not identified as a threat (miss) resulted in a lower performance score. More details about the task environment are described in (van Maanen et al., 2008). The second task (gauge task) was to monitor the temperature of the radar displayed on a meter (Figure 2). If the temperature dropped below zero or exceeded 300, participants had to press the control key to reset the meter. Resetting the meter either too soon or too late resulted in a lower score.

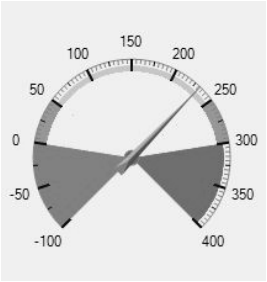


Figure 2. Meter (gauge) that displays the temperature of the radar.

2.3 Design

A 3 (model type) $\times$ 2 (task complexity) $\times$ 2 (performance level) design was used. Task complexity is a within-subjects factor and the order was balanced between the participants. Performance level was a quasi-independent variable we used to categorize participants.

2.4 Independent Variables

2.4.1 Type of Cognitive Model of Attention: The gaze-based model uses gaze behavior to estimate where attention is allocated. Eye-movements of the participants were recorded with an eye-tracker. The task-based model uses information about how the task is expected to be performed. The combined model uses both gaze behavior and the task-based model to determine where attention is allocated (Bosse et al., 2009).

2.4.2 Complexity of the Task: A complex and simple scenario were developed by manipulating the ambiguity and the dynamics of the scenario of the tactical picture compilation task. Concerning ambiguity, with small differences in the threat level of contacts it will more difficult to identify the five most threatening contacts. Dynamics was manipulated by varying the number of threat level changes of contacts over time. With many changes in the threat level it will be difficult to identify the five most threatening contacts, because the number of times that the contacts need to be re-evaluated increases.

2.4.3 Performance Level: After the experiment, a median split was performed to separate good and poor performers.

2.5 Dependent Variables

2.5.1 Task Performance: The performance on the tactical picture compilation task was determined by the accuracy of the identification of the five most threatening contacts during the task. The performance on the gauge task was determined by the accuracy of resetting the meter.

2.5.2 Accuracy of the Models: At random moments, varying from 2–6 minutes, both tasks were frozen. During a freeze, participants had to select the contacts by clicking on those that received their attention in the past 3 seconds. None of the contacts had to be selected when attention was only allocated to the temperature meter. The selected contacts were matched with those predicted by the models. The accuracy of each model was determined by calculating the area under the ROC curve (AUC) per model (Swets, 1988). The AUC indicates how accurate a model is to describe the participant's dynamics of attention allocation.

2.6 Procedure

Both tasks were explained thoroughly to the participants before the experiment started. The criteria that had to be used for the tactical picture compilation task were explained using different examples. All participants were tested on whether they were able to correctly apply the criteria: when the score was below 80%, they received extra instructions and another test. Participants performed a practice trial in which they had to perform both tasks. It was stressed that both tasks were equally important and that both tasks had to be performed well to attain a good performance overall.

3 Results

3.1 Manipulation Check

The difference in task complexity between a simple and complex scenario was determined by measuring the performance on the tactical picture compilation task. The performance in the simple scenario was significantly higher than the performance in the complex scenario ($t(41) = 4.56$, $p < 0.01$). A median split was performed to the divide the group in good and poor performers. A t-test showed that the difference in performance between these groups was significant ($t(40) = -23.13$, $p < 0.01$).

3.2 Main Effects

A three-way ANOVA with planned comparisons was performed to test the hypotheses. A significant main effect was found for the model type ($F(2, 39) = 8.97$, $p < .01$), but not for task complexity ($F(1, 40) = 0.29$, $p = .59$) and task performance ($F(1, 40) = 1.35$, $p = .25$).

3.3 Effect of Model Type

The results of the paired right-tailed t-tests for hypothesis 6 are shown in Table I. As was expected, the AUC of the combined model was significantly higher than that of the task-based and the gaze-based model. Hypothesis 6 is therefore accepted.

Table I
T-TESTS FOR HYPOTHESIS 6.

Hyp.	Descr.	$M_1(SD_1)$	$M_2(SD_2)$	t	df	p
$H6$	$C > G$	.66 (.08)	.58 (.06)	4.98	82	.00*
	$C > T$	.66 (.08)	.62 (.08)	2.19	82	.02*

Note. C = combined; G = gaze-based; T = task-based model
* $p < .05$

3.4 Effect of Task Complexity

Figure 3 displays the differences between the mean AUC for all three models in the simple and complex conditions. The results of the paired right-tailed t-tests for hypotheses 1, 2 and 3 are shown in Table II.

For all three models there were no significant differences in AUC between the simple and complex condition. Hypothesis 1 is therefore rejected. The AUC of the combined model was higher than that of the other models in both the simple and complex condition. However, not all differences were significant. In the simple condition, the difference between the combined model and the task-based model was not significant. Hypothesis 2 is therefore not fully accepted. Furthermore, it was expected that the difference between the combined model and the other models is higher in the complex condition than in the simple condition. This difference was not significant. Hypothesis 3 is not accepted.

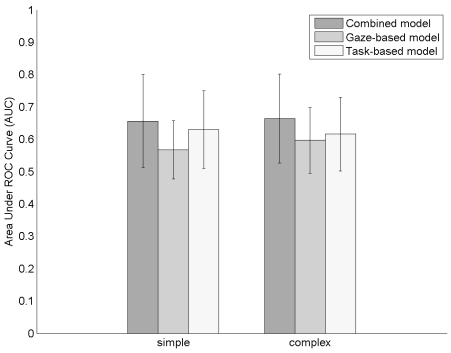


Figure 3. Model performance (AUC): simple and complex condition.

Table II
T-TESTS FOR HYPOTHESES 1, 2 AND 3.

Hyp.	Descr.	$M_1(SD_1)$	$M_2(SD_2)$	t	df	p
$H1$	$Cs > Cc$	.66 (.14)	.66 (.14)	$-.28$	82	.61
	$Gs > Gc$	.57 (.09)	.57 (.09)	-1.43	82	.92
	$Ts > Tc$	.63 (.12)	.62 (.11)	.54	82	.30
$H2$	$Cs > Gs$	.66 (.14)	.57 (.09)	3.40	82	.00*
	$Cs > Ts$	.66 (.14)	.63 (.12)	.91	82	.18
	$Cc > Gc$	.66 (.14)	.60 (.10)	2.57	82	.01*
	$Cc > Tc$	.66 (.14)	.62 (.11)	1.77	82	.04*
$H3$	$Cc - Gc > Cs - Gs$	.07 (.06)	.09 (.09)	-1.33	82	.91
	$Cc - Tc > Cs - Ts$	.05 (.09)	.03 (.06)	1.41	82	.08

Note. C = combined; G = gaze-based; T = task-based model, s = simple; c = complex condition
* $p < .05$

3.5 Effect of Task Performance

Figure 4 displays the differences between the mean AUC for all three models for good and poor performers. The results of the right-tailed t-tests for hypothesis 4 (unpaired) and 5 (paired) are shown in Table III.

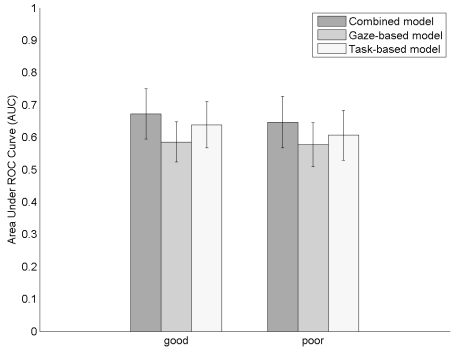


Figure 4. Model performance (AUC): good and poor performer condition.

Table III
T-TESTS FOR HYPOTHESES 4 AND 5.

Hyp.	Descr.	$M_1(SD_1)$	$M_2(SD_2)$	t	df	p
$H4$	$Cg > Cp$	.67 (.08)	.65 (.08)	1.03	40	.15
	$Tg > Tp$	.64 (.07)	.61 (.08)	1.41	40	.08
$H5$	$Cg > Gg$	.67 (.08)	.59 (.06)	3.96	40	.00*
	$Cg > Tg$	.67 (.08)	.64 (.07)	1.44	40	.08
	$Cp > Gp$	.65 (.08)	.68 (.07)	3.07	40	.00*
	$Cp > Tp$	.65 (.08)	.61 (.08)	1.67	40	.05

Note. C = combined; G = gaze-based; T = task-based model, g = good; p = poor performers condition.
* $p < .05$

Contrary to what was expected, no significant difference was found between good and poor performers for the combined and task-based model. Hypothesis 4 is not accepted. The AUC of the combined model is higher than the AUC of the other two models for both good and poor performers. The difference between the combined and gaze-based model was significant. Hypothesis 5 is therefore partly accepted.

4 CONCLUSION AND DISCUSSION

In this experiment participants had to execute a naval tactical picture compilation task. During the task participants had to indicate to what set of objects (ships) on the screen their attention was allocated. This was compared with the estimates of three cognitive models of attention: a gaze-based model, a task-based model and a combined model. Model performance was calculated for simple and complex task scenarios and for good and poor performers.

Results show that overall the combined model was a significant better predictor of attention allocation than the gaze-based and the task-based model. For both simple and complex task scenarios and for both good and poor performers the performance of the combined model was better than the performance of the other models. However, these differences were not always significant. In the complex scenario, the combined model was significantly better than both models. In the simple scenario the combined model was only significantly better than the gaze-based model. It seems that in simple tasks the inclusion of gaze-based information to the task-based model does not result in significantly more predictive power. This might be explained by that the complexity of models, e.g., single versus combined models, only has a positive effect when it is applied in complex scenarios. For good and poor performers the combined model was significantly better than the gaze-based model, but not significantly better than the task-based model. Our results indicate that for both good and poor performers, the inclusion of gaze-based information to the task-based model does not result in significantly more predictive power.

We did not find that the models were consistently better in the simple scenario compared to the complex scenario. We also did not find that the combined model and the task-based model were better for good performers than for poor performers. The difference in task complexity and performance level or the number of participants may have been insufficient to fully confirm our hypotheses. Even though the manipulation check showed significant differences in performance of the participants between both conditions, these differences might not

have been enough to cause an effect in the performance of the models.

Although it is too early to tell whether the predictions of the combined model are reliable enough for application in support systems, our results suggest that a combined model is the best candidate for this in terms of robustness and performance. To enhance the performance of the models, optimal parameter values need to be determined. Further, more research is needed to determine whether the predictive power of the models can be improved by adding components, such as expertise of the user or the visual saliency of information. Such an investigation can be done using similar validation techniques as described in this paper. Furthermore, since the AUC performance measure is decision criterion-independent, it needs to be determined whether liberal or conservative criterion settings are more effective for the prediction of human attention allocation or whether this criterion should be determined dynamically. Finally, in the near future experiments in which participants perform a tactical picture compilation task with and without support based on a combined model are needed as a more objective means for model evaluation.

Acknowledgments

This research was funded by the Dutch Ministry of Defence under program number V524. The authors would like to thank Jan Maarten Schraagen for his helpful comments.

References

Bosse, T., van Maanen, P.-P., and Treur, J. (2009). Simulation and formal analysis of visual attention. *Web Intelligence and Agent Systems: An International Journal (WIAS)*, 7(1):89–105.

Dzindolet, M. T., Peterson, S. A., Pomransky, R. A., Pierce, L. G., and Beck, H. P. (2003). The role of trust in automation reliance. *International Journal of Human Computer Studies*, 58(6):697–718.

Green, C. S. and Bavelier, D. (2003). Action video game modifies visual selective attention. *Nature*, 423:534–537.

Grootjen, M. and Neerincx, M. (2005). Operator load management during task execution in process control. In *Human Factors Impact on Ship Design*.

Just, M. and Carpenter, P. (1976). Eye fixations and cognitive processes. *Cognit ve Psychology*, 8:441–480.

Kahneman, D. (1973). *Attention and effort*. Prentice Hall, Englewoods Cliffs, NJ.

Ouerhani, N., Von Wartburg, R., Hügli, H., and Müri, R. (2004). Emperical validation of the saliency-based model of visual attention. *Electronic Letters on Computer Vision and Image Analysis*, 3(1):13–14.

Parasuraman, R. and Riley, V. A. (1997). Humans and automation: Use, misuse, disuse, abuse. *Human Factors*, 39:230–253.

Pavel, M., Wang, G., Li, K., and Li., K. (2003). Augmented cognition: Allocation of attention. In *Proceedings of 36th Hawaii International Conference on System Sciences*, pages 286–300. IEEE Computer Society.

Posner, M. I. (1980). Orienting of attention. *Quarterly Journal of Experimental Psychology*, 32:325.

Salvucci, D. D. (2000). A model of eye movements and visual attention. In *Third International Conference on Cognitive modeling*, pages 252–259.

Swets, J. A. (1988). Measuring the accuracy of diagnostic systems. *Science*, 240:1285–1293.

Treisman and Galade (1980). A feature-integration theory of attention. *Cognitive Psychology*, 12:97–136.

van Maanen, P.-P., de Koning, L., and van Dongen, K. (2008). Design and validation of habta: Human attention-based task allocator. In Mühlhäuser, M., Ferscha, A., and Aitenbichler, E., editors, *Proceedings of the First International Workshop on Human Aspects in Ambient Intelligence*, volume 11 of *Communications in Computer and Information Science (CCIS)*, pages 286–300. Springer-Verlag.

Wickens, C. D. (1984). Processing resources in attention. In Parasuraman, R. and Davies, D. R., editors, *Variaties of attention*, pages 63–101, Orland, FL. Academic Press.

Wickens, C. D. and McCarley, J. S. (2007). *Applied attention theory*. CRC Press, Boca Raton, FL.

Wood, R. E. (1986). Task complexity: definition of construct. *Organizational Behavior and Human Decision Processes*, 37:60–82.

Chapter 13

Personalization of Computational Models of Attention by Simulated Annealing Parameter Tuning

Apart from a few improvements, this chapter appeared as (van Lambalgen and van Maanen, 2010).

Personalization of Computational Models of Attention by Simulated Annealing Parameter Tuning

Rianne van Lambalgen* and Peter-Paul van Maanen*†

* Department of Artificial Intelligence, Vrije Universiteit Amsterdam
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
Email: rm.van.lambalgen@cs.vu.nl
† Department of Cognitive Systems Engineering, TNO Human Factors
P.O. Box 23, 3769 ZG Soesterberg, The Netherlands
Email: peter-paul.vanmaanen@tno.nl

***Abstract*—In this paper it is explored whether personalization of an existing computational model of attention can increase the model's validity. Computational models of attention are for instance applied in attention allocation support systems and can benefit from this increased validity. Personalization is done by tuning the model's parameters during a training phase, using Simulated Annealing (SA). The adapted attention model is validated using a task, varying in difficulty and attentional demand. Results show that the attention model with personalization results in a more accurate estimation of an individual's attention as compared to the model without personalization.**

***Index Terms*—Attention Model, Personalization, Parameter Tuning.**

1 Introduction

In a critical domain such as that of Naval Warfare, it is important for the crew to be aware of the situation on the field. However, the person has to deal with a large number of tasks in parallel and often the radar contacts are simply too numerous and dynamic to be adequately monitored by a single human. In (Both et al., 2009a), a simulation-based environment is presented that is similar to this Naval domain. In this environment, a variable amount of contacts have to be monitored, identified and handled.

Since attention is typically directed to one bit of information at a time (Treisman, 1988; Posner, 1980), a supporting software agent can be used. The agent alerts the human about a contact if it is ignored. To this end the agent has to maintain a model of the cognitive state of the human including the human's distribution of attention. In (Bosse et al., 2009), a validation is done on such a support agent by using a task called the Tactile Picture Compilation Task. It is shown that the designed support agent indeed improves the human's performance.

The existing model of the attention distribution (see also Bosse et al., 2007) is static in the sense that parameter values are set beforehand. However, it is known that such parameters may depend on personal characteristics and therefore it is useful to adjust them for each person performing a task. The focus of this research is to personalize the attention model by using Simulated Annealing (SA) to tune these parameters. Earlier work on validation of a cognitive model shows that appropriate parameter values can be found using SA (Both et al., 2009b).

Personalization of the attention model is done by tuning the model's parameters during a training phase. For validation of the adapted attention model, an experiment is conducted in the simulation-based environment. The task participants had to perform varied in difficulty and attentional demand. To obtain results on a person's attention, participants had to indicate the objects that had their attention at a certain moment in time.

In Section 2, the existing attention model is shown and a theoretical background is given on individual differences in attention. In Section 3 the task and procedure of the experiment are described, followed by a description of the applied SA algorithm in Section 4. Next, Section 5 compares the validity of the attention model with personalization to that

of the attention model without personalization. The results are discussed in Section 6.

2 Attention Model

2.1 Description of the Attention Model

The Attention Model is taken from (Bosse et al., 2009) and is briefly summarized in this section. The Attention Model is a mathematical model that uses input from features of objects on the screen and an agent's gaze to provide an estimation of the current attention distribution at a time point: an assignment of attention values $AV(s,t)$ to a set of attention spaces (i.e., areas on a computer screen) at that time. The attention distribution is assumed to have a certain persistency. At each point in time the new attention is related to the previous attention:

$$\begin{aligned} AV(s,t) &= \lambda \cdot AV(s,t-1) \\ &+(1-\lambda) \cdot AV_{norm}(s,t) \end{aligned} \quad (1)$$

where λ is the decay parameter for the decay of the attention value of space s at time point $t-1$, and $AV_{norm}(s,t)$ is determined by normalization for the total amount of attention, described by:

$$AV_{norm}(s,t) = \frac{AV_{new}(s,t)}{\sum_{s'} AV_{new}(s',t)} \cdot A(t) \quad (2)$$

$$AV_{new}(s,t) = \frac{AV_{pot}(s,t)}{1+\alpha \cdot r(s,t)^2} \quad (3)$$

Here $AV_{new}(s,t)$ is calculated from the potential attention value of space s at time point t and the relative distance of each space s to the gaze point (the center). The term $r(s,t)$ is taken as the Euclidean distance between the current gaze point and s at time point t (multiplied by an importance factor α which determines the relative impact of the distance to the gaze point on the attentional state, which can be different per individual and situation):

$$r(s,t) = d_{eucl}(gaze(t), s) \quad (4)$$

The potential attention value $AV_{pot}(s,t)$ is based on the features of the space (i.e., of the types of objects present) at that time (e.g., luminance, color):

$$AV_{pot}(s,t) = \sum_{maps\ M} M(s,t) \cdot w_M(s,t) \quad (5)$$

For every feature there is a saliency map M, which describes its potency of drawing attention (e.g., Itti and Koch, 2001). Moreover, $M(s,t)$ is the unweighted potential attention value of s at time point t, and $w_M(s,t)$ is the weight used for saliency map M, where $1 \leq M(s,t)$ and $0 \leq w_M(s,t) \leq 1$.

2.2 Individual Differences in Attention

Previous literature shows that individual differences in working memory capacity and general intelligence result in differences in controlled attention (Kane and Engle, 2002; Barrett et al., 2004). Controlled attention (i.e., top-down) can be seen as the ability to focus attention and the ability to prevent it to be captured by other events (mental or environmental). This indicates that for individuals with high working memory capacity, features that involve top-down attention will attract more attention as compared to features that automatically capture attention (e.g., luminance, color). As a consequence, those individuals will show less switching of attention from one location to the other. In the attention model, this could indicate a lower value for the decay (λ) of attention at a location.

In addition, it is known that factors like exhaustion and experienced pressure (i.e., a human's functional state (Both et al., 2009b)) influence the human performance and attention (Hockey, 2003). For example, the experienced pressure of a person can cause tunnel vision (narrowing of the attentional field), which has effect on α; the impact of the distance from the gaze point to a location on the attention value at that location. Since the functional state can differ across individuals, these factors should be taken into account in the estimation of attention.

Considering these differences, the attention model should be adjusted for each individual. In order to obtain a personalized model, in this paper parameter tuning is performed to a number of parameters described in the attention model in Section 2.2. As explained in this section, the decay parameter can depend on the individual and will therefore be tuned.

Furthermore, parameters are tuned that concern the weight from the saliency map of a feature to the potential attention value. Values of these weight variables may differ between individuals with low

versus high memory capacity. The exact parameters that are tuned to obtain a personalized attention model, depend on the specific task and are described in the parameter tuning section.

3 Method

3.1 Simulation-based Training Environment

The Simulation-based training Environment (Both et al., 2009a) that was used in this study consists of identifying incoming contacts and, based on the outcome of identification, deciding to eliminate the contact (by shooting) or allowing it to land (by not shooting). The participant controls a (stationary) weapon located at the bottom of a computer screen. In addition, contacts (allies and enemies in the shape of a dot) appear at a random location on the top of the screen and fall down to a random location at the bottom of the screen. A screen-shot of the environment is shown in Figure 1.

Before a contact can be identified, it has to be perceived. This is done by a mouse click at the contact, which reveals a mathematical equation underneath the contact. The identification task is to check the correctness of the mathematical equation (which is less difficult in less demanding situations). A correct equation means that the contact is an ally; an incorrect equation indicates that the contact is an enemy. Identification is done by pressing either the left or right arrow for respectively an ally or enemy. When a contact is identified, a green (for an ally) or a red (for an enemy) circle appears around the contact.

The contacts that have been identified as an enemy have to be shot before they land. A missile is fired by executing a mouse click at a specific location; the missile will move from the weapon to that location and explode exactly at the location of the mouse click. When a contact is within a radius of 50 pixels of the exploding missile, it is destroyed.

In this scenario participants have to pay attention to the most important contacts on the screen for an optimal performance of the task. However, when the number of contacts is large, this is not always possible. In addition, the task is also cognitively challenging, given the mathematical equations that have to be classified as either correct or incorrect. These characteristics are similar to real-life situations (e.g., air traffic control), which allows us to test the attention model in a situation close to reality.

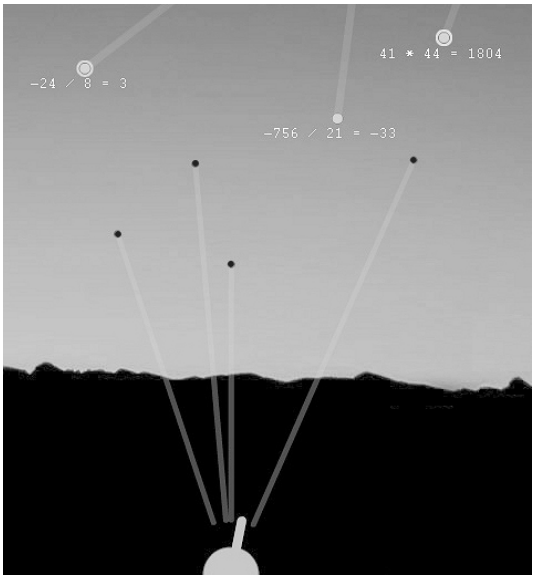


Figure 1. Screen-shot of the simulation-based training environment.

3.2 Participants and Procedure

In this study, 2 female participants and 3 male participants with a mean age of 22.8 took part. All participants already had some experience with the experimental environment.

The experiment consisted of 2 blocks of 20 minutes of the Experimental Task. In the first 10 minutes of one block, task demands were low (contacts appear every 10 to 20 seconds) and in the second 10 minutes of one block, task demands were high (contacts appear every 2.25 to 4.5 seconds). Furthermore, in the condition with high demands, mathematical equations were more difficult than in the condition with low demands.

In both blocks, 'freezes' were made after each 2.5 minutes. When a freeze was made, the experiment was put on hold and the following sentence was shown: "Gameplay frozen. Select contacts, press space when done." At this moment, participants had to select all contacts which they thought to have recently paid attention to. After selection,

a computer version of the NASA-TLX was shown, where participants had to indicate their performance and mental effort.

Throughout the experiment, eye gaze was measured using a Tobii x60 eye tracker. For this, calibration was performed at the start of the experiment. When eye tracking was optimal, on-screen instructions were given on the task environment and freezes. The instructions were followed by a practice block of two minutes medium task demands to get familiar with the environment. After practice, participants started with the first block. After the first block, the participant was given a three minute break before continuing with the next block.

4 Personalization of Computational Models of Attention

For data analysis, predictions of the two attention models (with and without personalization) were checked and compared for validation. The parameters that are tuned in the attention model are listed in Section 4.1 and a description of the procedure to personalize attention models is given in Section 4.2. The evaluation procedure used to check the models for validity is described in Section 4.3, and the eventual data analyses is described in Section 4.4.

4.1 Parameters to be Tuned

In Table I, an overview is given of the parameters that are tuned for personalization. Note that the weight values represent the weight of a feature from a saliency map to the potential attention value of that feature (equation 5 of the attention model).

Table I
Overview of the tuned parameters.

Parameter	Description
speed weight	weight of speed of an object (w_S in equation 5)
distance weight	weight of distance of an object (w_D in equation 5)
friend weight	weight of the identity of an object (friendly or hostile) (w_F in equation 5)
gaze weight	weight of a person's gaze at a location (α in equation 3)
decay factor	factor of keeping attention at a location (λ in equation 1) location
task factor	task influence when there are no objects in an attention space
amount of ms before freeze	the time a person takes into account when asked which contacts 'recently' got their attention.

Table II
Bounds of model parameters.

Parameter	Upper Bound	Lower Bound
speed weight	0	2
distance weight	0	2
friend weight	0	2
gaze weight	0	140
decay factor	0.8	1
task factor	0	1
amount of ms before freeze	1500	3500

4.2 Simulated Annealing Parameter Tuning

For personalizing the attention model for each participant, Simulated Annealing was used. This method uses a probabilistic technique to find a parameter setting (see description of the parameters of the attention model in Section 4.1). The parameter setting of the model without personalization is chosen as the best available parameter setting at the start. These settings were proven to be properly set by hand in different pilots.

After the initial phase a replacement is introduced into these settings to generate a neighbor of the current parameter settings in the search space. To limit the search space, upper and lower bounds were introduced for each parameter (see Table II). This leads to the following code (in Matlab language):

```
neighbor(x, bounds, jumpfactor) = X +
times(randperm(length(X)) == length(X),
(bounds(:,2) - bounds(:,1))' ) *
randn * permfactor
```

where `X` is a vector of the current parameter setting, times is vector multiplication, `randperm(n)` returns a random permutation of integers until `n`, `bounds(:,1)` and `bounds(:,2)` are the lower and upper bounds, `randn` is a random number between 0 and 1, and `permfactor` is the permutation speed.

If the neighbor is found to result in a higher validity (i.e., a lower energy value E) of the model (described later in Section 4.3) then it is marked

Algorithm 1 SA-PARAMETER-TUNING(X, C_{max}, B, T_{min}, T, S_{max}, R_{max}, b, j)

1: $X_{best} \leftarrow X$, $X_{select} \leftarrow X$, $C \leftarrow 1$, $S \leftarrow 0$, $E \leftarrow 1$, $E_{best} \leftarrow 1$, $R \leftarrow 0$
2: **while** $C_{max} \leq C$ **and** $T \leq T_{min}$ **and** $S_{max} \geq S$ **and** $R_{max} \geq R$ **do**
3: **while** X not changed or not within bounds **do**
4: $X_{new} \leftarrow$ NEIGHBOR(X_{select}, b, j)
5: $E_{new} \leftarrow 1 -$ AUC-EVALUATE(X_{new}, B)
6: **end while**
7: **if** $E_{new} < E_{best}$ **then**
8: $X_{best} \leftarrow X_{new}$
9: $E_{best} \leftarrow E_{new}$
10: **end if**
11: **if** $E_{new} < E_{selected}$ **then**
12: $X_{selected} \leftarrow X_{new}$
13: $E_{selected} \leftarrow E_{new}$
14: $S \leftarrow S + 1$
15: $R \leftarrow 0$
16: **else if** random $< e^{(E_{selected} - E_{new})/T}$ **then**
17: $X_{selected} \leftarrow X_{new}$
18: $E_{selected} \leftarrow E_{new}$
19: $S \leftarrow S + 1$
20: **else**
21: $R \leftarrow R + 1$
22: **end if**
23: DECREASE(T) {for instance $T \leftarrow 0.8 \cdot T$}
24: $C \leftarrow C + 1$
25: **end while**
26: **return** X_{best}

as the best known parameter setting. Otherwise the neighbor can still be selected with a small probability dependent on a decreasing temperature value. The best parameter setting is always stored, also when a different neighbor is selected. When a neighbor is not selected a new neighbor is generated to evaluate its appropriateness, and so on, until certain stopping criteria hold. Eventually the set of best parameters converges to the optimal solution (i.e., due to the probabilistic character of SA, the found solution is semi-optimal). The pseudo-code of this procedure is found in Algorithm 1.

The input variables of Algorithm 1 are the initial parameter vector X, maximum computation steps C_{max}, observed human behavior B, minimum temperature T_{min}, initial temperature T, maximum successes S_{max}, maximum consecutive rejections R_{max}, bounds b and jump factor j. The output variable is the best estimate of parameter settings X_{best}. This algorithm has been implemented using Matlab.[1].

Table III
CONFUSION MATRIX.

		Participant		
		t	f	*total*
Model	t'	Hits	False Alarms	T'
	f'	Misses	Correct Rejections	F'
	total	T	F	

4.3 Subjective Evaluation Measure

The validity was measured based on a subjective measure of attention (in terms of personal reports by the participants during the freezes). The output of the models have been compared with subjective data retrieved during the freezes in the experiment. This means that both the models as well as the participants indicated where the attention of the participant was allocated to, and these indications were compared with each other afterwards.

The comparison of the models with the subjective data was done using Area Under the Curve (AUC) analysis. AUCs are solutions of the integrals of the Receiver-Operator Characteristic (ROC) curves. ROC curve analysis was adopted from signal detection theory, where the sensitivity and specificity of detecting a signal, i.e., in our case the detection of the fact that the participant was paying attention to an object, is quantified and visualized. The ROC curves can be drawn by plotting sensitivity (hit rate) versus $1-$ specificity (false alarm rate). The hit rate and false alarm rate can be extracted from confusion matrices, as is shown in Table III.

In Table III, t and f represent whether the participant indicated that he allocated attention to an object or not, respectively, and t' and f' indicate that one of the models indicated it or not, respectively. For drawing the ROC-curve, Hits$/T$ results in the

[1]The used Matlab scripts, different visualizations of the data and results can be found at `http://www.few.vu.nl~pp/attention/ECAI10`.

Algorithm 2 AUC-EVALUATE(X, B)

1: **for all** decision thresholds th **do**
2: **for all** freezes fr **do**
3: Calculate whether there is a HIT, MISS, FA or CR, using parameters X for calculating the attention value AV during freeze fr (in Table III: if $AV > th$, then t' and otherwise f', and if $B == 1$, then t and otherwise f)
4: **end for**
5: Construct confusion matrix
6: **end for**
7: Plot ROC-curve using all hit and false alarm rates in confusion matrices
8: Calculate AUC (where $AUC \approx 1$ is good and $AUC \approx .5$ is poor performance)
9: **return** AUC

hit rate and False Alarms/F results in the false alarm rate.

The speudo-code of the procedure of testing the hypotheses based on the subjective data is found in Algorithm 2.

The input variables of Algorithm 2 are the parameter vector X and the observed human behavior B. The output variable is the area under the curve AUC. The above procedure has also been implemented using Matlab.

4.4 Data Analysis

In order to determine whether personalization is beneficial in terms of validity, for each participant the 5 AUCs per condition for each model type is compared with each other, using a paired one-sided t-test. The model without personalization used fixed parameters and the model with personalization used the for each participant tuned parameters.

5 Results

The personalization procedure has been applied to all participants as mentioned in Section 3.2. The result is shown in Figure 2 where the energy of the best parameter settings for each participant is set out against the number of computation steps. As you can see, the results converge to the optimum (zero energy).

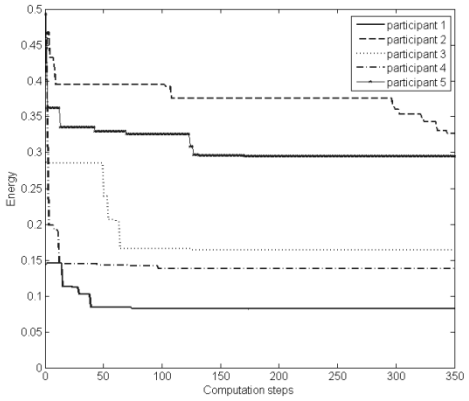


Figure 2. Best energy values per participant during SA parameter tuning.

Figure 3 shows the mean AUCs for the model with and without personalization. A paired one-sided t-test showed that the personalized attention models have a significantly mean AUC compared to the fixed attention models ($t(8) = 2.00$, $p = 0.042$). Hence, based on the used subjective validation data, the hypothesis that personalization of attention models indeed can be accepted.

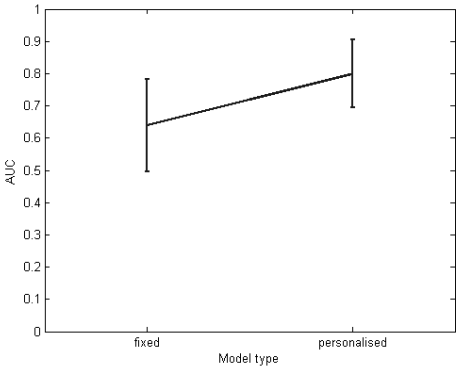


Figure 3. Means area under the curve (AUC) for the model without (left) and with (right) personalization.

6 Discussion

In this paper, a previously presented attention model has been applied on a dynamic task, varying in attentional demands. In addition, personalization, using Simulated Annealing parameter tuning, has been applied on the model. Validation results showed a significant difference between the personalized attention model and the fixed attention model, indicating that the former (personalized) was better in predicting attention as compared to the latter (fixed).

The validation was subjective in the sense that a participant's own estimation was measured by asking to which objects they had directed their attention before the given freezes. The subjective measure makes the assumption that people have good introspection skills, i.e., that they are accurate in the estimation of their own attention distribution. Although various experiments have pointed out that this is a reasonable assumption, it makes sense to analyze the results using an objective measure as well. A possible way of measuring objective attention is by looking at mouse clicks at a location. In future research, this can be taken into account, in addition to the subjective measure.

Simulated Annealing parameter tuning has proven to be effective in estimating a person's attention in this specific task. The same algorithm is previously used to obtain a personalized model for a Human's Functional State (Both et al., 2009b). The application to two different models shows that the SA algorithm can be used for parameter tuning for different situations and different models. However, it should be noted that SA is a probabilistic procedure and therefore is suboptimal, specifically as the necessary computing capacity becomes relatively smaller compared to the problem space.

In the literature, personalization is commonly used, but mainly in the area of human-computer interaction to obtain a user model for a personalized user interface (Henricksen and Indulska, 2005). Also, in previous research, neural networks are used to estimate the operator functional state (Wilson and Russell, 2003). However, although attention models have previously been proposed (Kane and Engle, 2002; Chen et al., 2003), the authors know of no attention model that has been adjusted to the individual.

In the future personalization of attention models can be extended. In the current personalized model parameters are tuned that are known to differ per individual. However, in future research personalization can be done by using collected data on personality to improve the attention model. Furthermore, in the current personalized model, parameters like the attention threshold and the total amount of attention are static. These could be coupled to a individual's functional state (e.g., experienced pressure, exhaustion), making the model fit for each individual, but also in different conditions (high/low workload). Such adjustments are expected to result in again an increase of the model's validity.

Acknowledgments

The authors would like to thank Jan Treur, Tibor Bosse and Alexei Sharpanskykh for their support and helpful comments. Special acknowledgments go to Rogier Oorburg and Michael Vos for implementing the Simulation-based Training Environment.

References

Barrett, L., Tugade, M., and Engle, R. (2004). Individual differences in working memory capacity and dual-process theories of mind. *Psychological Bulletin*, 130(4):553–573.

Bosse, T., van Lambalgen, R., van Maanen, P.-P., and Treur, J. (2009). Attention manipulation for naval tactical picture compilation. In *Proceedings of the 2009 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'09)*.

Bosse, T., van Maanen, P.-P., and Treur, J. (2007). Simulation and formal analysis of visual attention in cognitive systems. In *Proceedings of the Fourth International Workshop on Attention in Cognitive Systems (WAPCV'07)*. Published as: L. Paletta, E. Rome (Eds.), Attention in Cognitive Systems, Lecture Notes in AI, Springer Verlag, 2007.

Both, F., Hoogendoorn, M., van Lambalgen, R., Oorburg, R., and de Vos, M. (2009a). Relating personality and physiological measurements to task performance quality. In Taatgen, N. A. and van Rijn, H., editors, *Proceedings of the 31th Annual Conference of the Cognitive Science Society (CogSci'09)*, pages 2819–2825.

Both, F., Hoogendoorn, M., W., J. S., van Lambalgen, R., Oorburg, R., Sharpanskykh, A., Treur, J., and de Vos, M. (2009b). Adaptation and validation of an agent model of functional state and performance for individuals. In *Proceedings of 12th International Conference on Principles of Practice in Multi-Agent Systems (PRIMA'09)*.

Chen, L., Xie, X., Fan, X., Ma, W., Zhang, H., and Zhou, H. (2003). A visual attention model for adapting images on small displays. *ACM Multimedia Systems Journal*.

Henricksen, K. and Indulska, J. (2005). Personalising context-aware applications. In Meersman, R., Tari, Z., and Herrero, P., editors, *On the Move to Meaningful Internet Systems 2005: OTM Workshops*, volume 3762 of *LNCS*, pages 122–131.

Hockey, G. (2003). Operator functional state as a framework for the assessment of performance degradation. In Hockey, G. R. J., Gaillard, A. W. K., and Burov, O., editors, *Operator Functional State*, pages 8–21. IOS Press.

Itti, L. and Koch, C. (2001). Computational modeling of visual attention. *Nature Reviews Neuroscience*, 2(3):194–203.

Kane, M. and Engle, R. (2002). The role of prefrontal cortex in working-memory capacity, executive attention, and general fluid intelligence: An individual-differences perspective. *Psychological Bulletin & Review*, 9(4):637–671.

Posner, M. I. (1980). Orienting of attention. *Quarterly Journal of Experimental Psychology*, 32:325.

Treisman, A. (1988). Features and objects: The 14th bartlett memorial lecture. *Q. J. Experimental Psychology A*, 40:201–237.

Wilson, G. and Russell, C. (2003). Operator functional state classification using multiple psychophysiological features in an air traffic control task. *Human Factors*, 45(3):381–389.

Chapter 14

A System to Support Attention Allocation: Development and Application

This chapter is partly based on (Bosse et al., 2009c,e,b). One of these papers (Bosse et al., 2009b) was awarded with the 2009 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2009) Best Paper Award. Also an extended abstract (Bosse et al., 2009d) appeared based on (Bosse et al., 2009c).

A System to Support Attention Allocation: Development and Application

Tibor Bosse*, Rianne van Lambalgen*, Peter-Paul van Maanen*† and Jan Treur*

* Department of Artificial Intelligence, Vrije Universiteit Amsterdam
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
Email: {tbosse, rm.van.lambalgen, treur}@cs.vu.nl
† Department of Cognitive Systems Engineering, TNO Human Factors
P.O. Box 23, 3769 ZG Soesterberg, The Netherlands
Email: peter-paul.vanmaanen@tno.nl

***Abstract*—This paper discusses and evaluates an agent model that is able to manipulate the visual attention of a human, in order to support naval crew. The agent model consists of four sub-models, including a model to reason about a subject's attention. The model was evaluated based on a practical case study which was formally analyzed and verified using automated checking tools. Results show how a human subject's attention is manipulated by adjusting luminance, based on assessment of the subject's attention. These first evaluations of the agent show a positive effect.**

***Index Terms*—Human Experimentation, Attention, Interface Agent, Formal Analysis.**

1 Introduction

In the domain of naval warfare, it is crucial for the crew of the vessels involved to be aware of the situation in the field. One of the crew members is usually assigned the task to identify and classify all entities in the environment (e.g., Heuvelink and Both, 2007). This task determines the type and intent of a multiplicity of contacts on a radar screen. Attention is typically directed to one bit of information at a time (Posner, 1980; Theeuwes, 1994; Treisman, 1988). A supporting software agent may alert the human about a contact if it is ignored. To this end the agent has to maintain a model of the cognitive state of the human including the human's distribution of attention. Existing cognitive models on attention show that it is possible to predict a person's attention based on a saliency map, calculated from features of a stimulus, like luminance, color and orientation (Itti and Koch, 2001; Parkurst et al., 2002). In this study, a Theory of Mind (or ToM, (e.g., Bosse et al., 2007b)) model is exploited within the agent model to analyze attention of the human. Attention can then be influenced (or 'manipulated') by changing features of stimuli, e.g., its contrast with stimuli at other locations (Itti and Koch, 2001; Levitt and Lund, 1997; Nothdurft, 2000), its luminance (Theeuwes, 1995; Turatto and Galfano, 2000), or its form (Turatto and Galfano, 2000).

Some approaches in the literature address the development of software agents with a Theory of Mind (e.g., Bosse et al., 2007b; Marsella et al., 2004; Memon and Treur, 2008), but only address a model of the epistemic (e.g., beliefs), motivational (e.g., desires, intentions), and/or emotional states of other agents. For the situation sketched above, attribution of attentional states has to be addressed. In the current paper, an agent model has been developed, which uses four specific (sub)models. The first is a representation of a dynamical model of human attention, for estimation of the locations of a person's attention, based on information about features of objects on the screen and the person's gaze. The second model is a reasoning model which the agent uses to reason through the first model, to generate beliefs on attentional states at any point in time. With a third model the agent compares the output of the second model with a normative attention distribution and determines the discrepancy. Finally, a fourth model is used to direct the person's attention to relevant contacts based on the output of

the third model.

Initial versions of the first two models were adopted from earlier work (Bosse et al., 2009c). The current paper focuses on the use of the last two models, where input from (Bosse et al., 2009b) was adopted. Section 2 gives a literature review on the manipulation of attention, Section 3 describes a formalization of the different models, and in Section 4 the global behavior of the model is tested by simulation experiments. In Section 5, the model is implemented in the context of a case study where a software agent is used to manipulate a subject's attention. Based on this case study, Section 6 addresses experimental validation of the results, and Section 7 addresses automated verification of different important properties of the sub-models used in the agent. In Section 8, a formal mathematical analysis of the model is given. Finally, Section 9 is a discussion.

2 Manipulation of Attention

Typically, a person's attention is influenced both by top-down and by bottom-up processes. The former means that observers orient their attention in a goal-directed manner, as a consequence of their expectations or intentions (Posner, 1980). For example, when searching for a friend in the crowd, attention is guided top-down (Theeuwes, 1994). In contrast, the latter means that attention is elicited by a (highly salient) trigger from the environment. For example, one green circle among several blue circles will "pop-out" and attention will be directed to this object (Treisman, 1988). In this project the focus is primarily on adjusting the features of a specific location, such that only bottom-up attention is manipulated. Features that are mainly known to influence attention are intensity (luminance), color and orientation. Previous research shows that attention can be elicited both by the contrast with stimuli at other locations (Itti and Koch, 2001; Levitt and Lund, 1997; Nothdurft, 2000) and the abrupt change of a feature, like luminance (Theeuwes, 1995; Turatto and Galfano, 2000) or form (Turatto and Galfano, 2000).

Several cognitive models on attention have been proposed and show that it is possible to predict attention allocation based on a saliency map, calculated from features of a stimulus, like luminance, color and orientation (Itti et al., 1998; Parkurst et al., 2002). Furthermore, other features like effort and expectancy have been incorporated in attention models (Gore et al., 2009; Wickens et al., 2008). These proposed models are not dynamic in the sense that they take changes of information from the environment into account. However, if indeed the change of a specific feature (like luminance) can cause an attention shift in the human performing a task considered, a *support model* can be used to realize this change. This way, humans who have to direct their attention to a large number of locations in parallel can be supported to adequately perform their task.

Although much is known on the features that guide attention (Itti and Koch, 2001; Theeuwes, 1994), there are few other attempts to design a system for attention allocation support. Automated attention guidance has been investigated, by providing either a tactical cue (Sklar and Sarter, 1999) or a visual cue to a relevant location (Horrey et al., 2006). However, this automated cueing is based on features of the task (i.e., threat of an object) and not on the human's actual distribution of attention.

3 A Theory of Mind for Attention

3.1 Overall Setting

A Theory of Mind enables an agent to analyze another agent's mind, and to act according to the outcomes of such an analysis and its own goals. For the general case such processes require some specific facilities.

A *representation of a dynamical model* is needed describing the relationships between different mental states of the other agent. Such a model may be based on qualitative causal relations, but it may also concern a numerical dynamical system model that includes quantitative relationships between the other agent's mental states. In general such a model does not cover all possible mental states of the other agent, but focuses on certain aspects, for example on beliefs and desires, on emotional states, on the other agent's awareness states, or on attentional states as in this paper.

Furthermore, *reasoning methods to generate beliefs on the other agent's mental state* are needed to draw conclusions based on the dynamical model in (1) and partial information about the other agent's

mental states. This may concern deductive-style reasoning methods performing forms of simulation based on known inputs to predict certain output, but also abductive-style methods reasoning from output of the model to (possible) inputs that would explain such output.

Moreover, when in one way or the other an estimation of the other agent's mental state has been found out, it has to be *assessed whether there are discrepancies* between this state and the agent's own goals. Here also the agent's self-interest comes in the play. It is analyzed in how far the other agent's mental state is in line with the agent's own goals, or whether a serious threat exists that the other agent will act against the agent's own goals.

Finally a *decision reasoning model* is needed to decide how to act on the basis of all of this information. Two types of approaches are possible. A first approach is to take the other agent's state for granted and prepare for the consequences to compensate for them as far as these are in conflict with the agent's own goals, and to cash them as far as they can contribute to the agent's own goals (*anticipation*). For the navy case, an example of anticipation is when it is found out that the other agent has no attention for a dangerous object, and it is decided that another colleague or computer system will handle it (dynamic task reallocation). A second approach is not to take the other agent's mental state for granted but to decide to try to get it adjusted by affecting the other agent, in order to obtain a mental state of the other agent that is more in line with the agent's own goals (*manipulation*). This is the case addressed in this paper.

The general pattern sketched above is applied in this paper to the way in which a (software) agent can attempt to adjust the other (human) agent's attention, whenever required. To this end the software agent uses the following four different models: Dynamic Attention Model, Model for Beliefs about Attention, Model to Determine Discrepancy and Decision Model for Attention Adjustment. In this section, each of these models are described in detail. The agent and its interaction with the environment (involving a complex task and an eye-tracker, see Section 5) are schematically displayed in Figure 1.

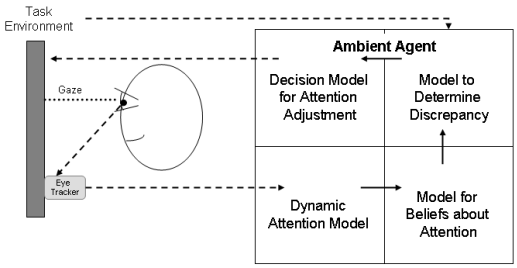


Figure 1. Overview of the ambient agent and its environment.

3.2 Dynamic Attention Model

This model is taken over from (Bosse et al., 2009c) and is only briefly summarized in this section. The model uses three types of input: information about the human's *gaze direction*, about *locations* (or spaces) and about *features* of objects on the screen. Based on this, it makes an estimation of the *current attention distribution* at a time point: an assignment of attention values $AV(s,t)$ to a set of attention spaces at that time. The attention distribution is assumed to have a certain persistency. At each point in time the new *attention level* is related to the previous attention, by:

$$\begin{aligned} AV(s,t) = \lambda \cdot AV(s,t-1) \\ +(1-\lambda) \cdot AV_{norm}(s,t) \end{aligned} \quad (1)$$

where λ is the decay parameter for the decay of the attention value of space s at time point $t-1$, and $AV_{norm}(s,t)$ is determined by normalization for the total amount of attention, described by:

$$AV_{norm}(s,t) = \frac{AV_{new}(s,t)}{\sum_{s'} AV_{new}(s',t)} \cdot A(t) \quad (2)$$

$$AV_{new}(s,t) = \frac{AV_{pot}(s,t)}{1+\alpha \cdot r(s,t)^2} \quad (3)$$

Here $AV_{new}(s,t)$ is calculated from the potential attention value of space s at time point t and the relative distance of each space s to the gaze point (the center). The term $r(s,t)$ is taken as the Euclidean distance between the current gaze point and s at time point t (multiplied by an importance factor α which determines the relative impact of the

distance to the gaze point on the attentional state, which can be different per individual and situation):

$$r(s,t) = d_{\mathit{eucl}}(\mathit{gaze}(t), s) \quad (4)$$

The potential attention value $AV_{\mathit{pot}}(s,t)$ is based on the features of the space (i.e., of the types of objects present) at that time (e.g., luminance, color):

$$AV_{\mathit{pot}}(s,t) = \sum_{\mathit{maps}\ M} M(s,t) \cdot w_M(s,t) \quad (5)$$

For every feature there is a saliency map M, which describes its potency of drawing attention (e.g., Chen et al., 2003; Itti et al., 1998; Itti and Koch, 2001). Moreover, $M(s,t)$ is the unweighted potential attention value of s at time point t, and $w_M(s,t)$ is the weight used for saliency map M, where $1 \leq M(s,t)$ and $0 \leq w_M(s,t) \leq 1$.

Figure 2 shows an overview of this model. The circles denote the italicized concepts introduced above, and the arrows indicate influences between concepts.

3.3 Model for Beliefs about Attention

This (reasoning) model is used to generate beliefs about attentional states of the other agent. The software agent uses the dynamical system model as described in Section 3.2 as an internal simulation model to generate new attentional states from the previous ones, gaze information and features of the object, with the use of a forward reasoning method (forward in time) as described in (Bosse et al., 2008). The basic specification of the reasoning model can be expressed by the representation leads_to_after(I, J, D) (belief that I leads to J after duration D). Here, I and J are both information elements (i.e., they may correspond to any concept from Figure 2, e.g., gaze_at(1, 2) or has_value(av(1,2), 0.68).

In addition, the representation at(I, T) gives information on the world (including human processes) at different points in time. It represents a belief that state I holds at time point T. For example, at(gaze_at(1,2), 53) expresses that at time point 53, the human's gaze is at the space with coordinates (1,2).

3.4 Model to Determine Discrepancy

With this model the agent determines the discrepancy between actual and desirable attentional states and to what extent the attention distribution has to change. This is based on a model for the desirable attention distribution (prescriptive model). For the case addressed this means an assessment of which objects deserve attention (based on features as distance, speed and direction). To be able to make such assessments, the agent is provided with some tactical domain knowledge, in terms of heuristics (also see Section 5).

3.5 Decision Model for Attention Adjustment

The model for adjustment of the attention distribution has as input the discrepancy determined by the model described in Section 3.4, and also makes use of the explicitly represented dynamical model as described in Section 3.2. The general idea is that the relations between variables within this model are followed in a backward manner, thereby propagating the desired adjustment from the attentional state variable to the features of the object at the screen. The general pattern behind this operation on a dynamical model representation is illustrated in Figure 3. Here v_1 is the (desired) output of a model, and by branches the variables on which this depends are depicted, until the leaves where actual adjustments can be made.[1]

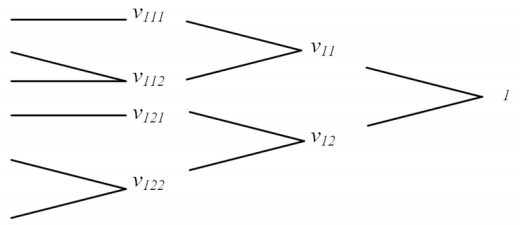


Figure 3. Dependencies between variables in a dynamical system model.

This is a form of desire refinement: starting from the root variable, by a step-by-step process a desire on adjusting a parent variable is refined to desires on

[1]For the moment, deterministic relationships between variables are assumed. However, in a later stage, the agent might learn such relationships.

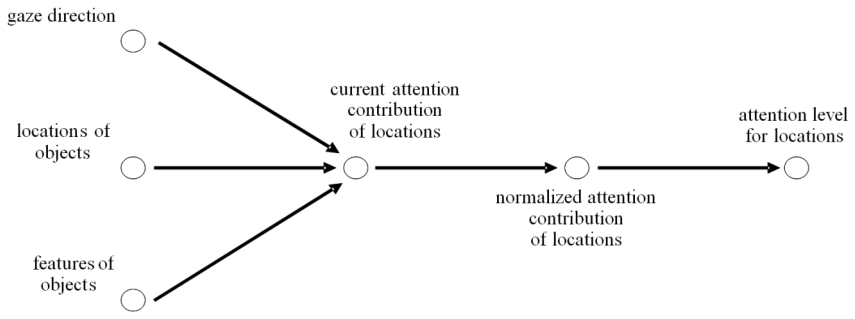


Figure 2. Overview of attention model.

adjustments of the children variables, until the leave variables are reached. The starting point is the desire on the root variable, which is the desired adjustment of the attentional state; this is determined by:

belief($av(s)<h$) $\wedge$ desire($a(v)>h$) $\wedge$
belief(has_value($av(s)$, v)) $\twoheadrightarrow$
desire(adjust_by($av(s)$, $(h-v)/v$)

Note that here the adjustment is taken relative (expressed by division of the difference $h-v$ by v). Suppose as a point of departure (given the discrepancy assessment) an adjustment Δv_1 is desired, and that v_1 depends on two variables v_{11} and v_{12} that are adjustable (the non-adjustable variables can be left out of consideration). Then by elementary calculus as a linear approximation the following relations between required adjustments can be obtained:

$$\Delta v_1 = \frac{\partial v_1}{\partial v_{11}} \Delta v_{11} + \frac{\partial v_1}{\partial v_{12}} \Delta v_{12}$$

This formula is used to determine the desired adjustments Δv_{11} and Δv_{12}, where by weight factors μ_{11} and μ_{12} the proportion can be indicated in which the variables should contribute to the adjustment:

$$\Delta v_{11}/\Delta v_{12} = \mu_{11}/\mu_{12}$$

$$\Delta v_1 = \frac{\partial v_1}{\partial v_{11}} \Delta v_{12} \mu_{11}/\mu_{12} + \frac{\partial v_1}{\partial v_{12}} \Delta v_{12} = (\frac{\partial v_1}{\partial v_{11}} \mu_{11}/\mu_{12} + \frac{\partial v_1}{\partial v_{12}}) \Delta v_{12}$$

So the adjustments can be made as follows:

$$\Delta v_{12} = \frac{\Delta v_1}{\frac{\partial v_1}{\partial v_{11}} \mu_{11}/\mu_{12} + \frac{\partial v_1}{\partial v_{12}}}$$

$$\Delta v_{11} = \mu_{11}/\mu_{12} \frac{\Delta v_1}{\frac{\partial v_1}{\partial v_{11}} \mu_{11}/\mu_{12} + \frac{\partial v_1}{\partial v_{12}}} = \frac{\Delta v_1}{\frac{\partial v_1}{\partial v_{11}} + \frac{\partial v_1}{\partial v_{12}} \mu_{12}/\mu_{11}}$$

Special cases are $\mu_{11} = \mu_{12} = 1$ (absolute equal contribution) or $\mu_{11} = v_{11}$ and $\mu_{12} = v_{12}$ (relative equal contribution: in proportion with their absolute values). As an example, consider a variable that is just the weighted sum of two other variables (as is the case, for example, for the aggregation of the effects of the features of the objects on the attentional state):

$$v_1 = w_{11} v_{11} + w_{12} v_{12}$$

For this case

$$\frac{\partial v_1}{\partial v_{11}} = w_{11} \qquad \frac{\partial v_1}{\partial v_{12}} = w_{12}$$

and

$$\Delta v_{11} = \frac{\Delta v_1}{w_{11} + w_{12} \mu_{12}/\mu_{11}}$$

$$\Delta v_{12} = \frac{\Delta v_1}{w_{11} \mu_{11}/\mu_{12} + w_{12}}$$

For example when $\mu_{11} = \mu_{12} = 1$ this results in

$$\Delta v_{11} = \frac{\Delta v_1}{w_{11} + w_{12}} \qquad \Delta v_{12} = \frac{\Delta v_1}{w_{11} + w_{12}}$$

Assuming $w_{11} + w_{12} = 1$ in addition, this results in $\Delta v_{11} = \Delta v_{12} = \Delta v_1$.

Another setting, which actually has been used in the model, is to take $\mu_{11} = v_{11}$ and $\mu_{12} = v_{12}$. In this case the adjustments are assigned proportionally; for example, when v_1 has to be adjusted by 5%, also the other two variables on which it depends need to contribute an adjustment of 5%. Thus the relative adjustment remains the same through propagations:

$$\frac{\Delta v_{11}}{v_{11}} = \frac{\Delta v_1}{w_{11} + w_{12}v_{12}/v_{11}}/v_{11} = \frac{\Delta v_1}{w_{11}v_{11} + w_{12}v_{12}} = \frac{\Delta v_1}{v_1}$$

This shows the general approach on how desired adjustments can be propagated in a backward manner through a dynamical model. Thus a desired adjustment of the attentional state as output at some point in time can be related to adjustments in the features of the displayed objects as inputs at previous points in time. For the case study undertaken this approach has been applied, although at some points in a simplified form. One of the simplifications made is that due to the linearity of most dependencies in the model, adjustments have been used that just propagate without any modification. An example of a rule specified to achieve this propagation process is:

desire(adjust_by(u_1, a)) $\wedge$ belief(depends_on(u_1, u_2)) $\twoheadrightarrow$ desire(adjust_by(u_2, a))

Here the adjustments are taken relative, so, this rule is based on $\Delta u_2/u_2 = \Delta u_1/u_1$ as derived above for the linear case. When at the end the leaves are reached, which is represented by the belief that they are directly adjustable, then from the desire an intention to adjust them is derived.

desire(adjust_by(u, a)) $\wedge$ belief(directly_adjustable(u)) $\twoheadrightarrow$ intention(adjust_by(u, a))

If an intention to adjust a variable u by a exists with current value b, the new value $b + \alpha * a *$ b to be assigned to u is determined; here α is a parameter that allows the modeler to tune the speed of adjustment:

intention(adjust_by(u, a)) $\wedge$ belief(has_value_for(u, b)) $\twoheadrightarrow$ performed(assign_new_value_for(u, $b + \alpha * a * b$))

This rule is applied for variables that describe features f of objects at locations s, i.e., instances for u of the form feature(s, f). Note that each time the adjustment is propagated as a value relative to the overall value.

4 Simulation Results

To test whether the approach described above yields the expected behavior, it has been used to perform a number of simulation experiments in the LEADSTO simulation environment (Bosse et al., 2007a). This environment takes a specification of causal relationships (in the format as shown in the previous sections) as input, and uses this to generate simulation traces. The simulations shown here address a slightly simplified case, where the radar screen has been split up in 4 locations. For the time being, it is assumed that each location contains one contact, and that these contacts stay within their locations.

The features of the contacts that are manipulated are luminance, size, and level of flashing. Initially, each contact starts with the same features, but during the simulation these features are manipulated, based on the prescribed (or desired) attention. This desired attention is generated randomly, where every 50 time units a next location is selected where the attention should be. Furthermore, the behavior of the human gaze is generated as follows: after each adaptation of the features, the gaze moves to one of the four locations, with a probability that is proportional to the saliency of the contact at that location.

The results of an example simulation run are depicted in Figures 4–7. In these figures, time is on the horizontal axis, and the different state of the process is shown in the vertical axis. A dark line indicates that a state is true at a certain time point. Note that some information has been omitted due to space limitations. Figure 4 shows the model-based reasoning process of the agent, in terms of desires

and intentions. Figures 5–7 show, respectively, the estimated attention, the human's gaze, and the value of the feature "luminance" at different locations over time. As shown in Figure 4, initially it is desired that at least 50% of the human's attention is at location 2 (desire($av(2)>0.5$)). Since this is not the case (see Figure 5), the luminance of the contact at location 2 is increased (see Figure 7). As a result, the human's gaze shifts towards this location (see Figure 6), which increases his attention for location 2. In the rest of the simulation, this pattern is repeated for different locations.

After successfully running simulations of the models under a number of different parameter settings, it was considered appropriate to be implemented in a real world case study. This case study is described in the next section.

5 Case Study

The different models have been implemented and tested for a case study. The used case study mimics a real-world situation, with human subjects executing the Tactical Picture Compilation Task. In Section 5.1 the environment is shortly explained. Section 5.2 discusses some implementation details of the attention manipulating agent tailored to the environment. In Section 5.3 the results are presented.

5.1 Environment

The task used for this case study is an altered version of the identification task described in (Heuvelink and Both, 2007) that has to be executed in order to build up a tactical picture of the situation, i.e., the Tactical Picture Compilation Task (TPCT). The implementation of the software was done in Gamemaker[2].

In Figure 8 a snapshot of the interface of the task environment is shown. The goal is to identify the five most threatening contacts (ships). In order to do this, participants monitor a radar display of contacts in the surrounding areas. To determine if a contact is a possible threat, different criteria have to be used. These criteria are the identification criteria (idcrits) that are also used in naval warfare, but are simplified in order to let naive participants learn them more easily. These simplified criteria are the speed (depicted by the length of the tail of a contact), direction (pointer in front of a contact), distance of a contact to the own ship (circular object), and whether the contact is in a sea lane or not (in or out the large open cross). Contacts can be identified as either a threat (diamond) or no threat (square).

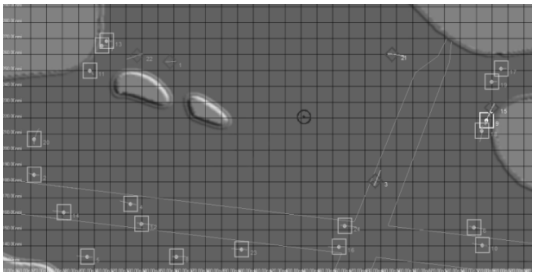


Figure 8. Interface of the task environment.

5.2 Implementation

The support agent was further developed and evaluated using Matlab (see appendix). The output of the environment described in Section 5.1 was used and consisted of a representation of all properties of the contacts visible on the screen, i.e., speed, direction, whether it is in a sea lane or not, distance to the own ship, location on the screen and contact number. In addition, data from a Tobii x50 eye-tracker[3] were retrieved from a participant executing the TPC task. All data were retrieved several times per second and were used as input for the models within the agent. Once the agent models were tailored to the TPC case study, the eventual implementation of them was done in C#.

In Figure 9 the interface of the implemented agent models is shown. This interface consists of four parts where parameters can be set. In first part the agent models can be run. Once the button is pushed, both the input from the TPC task environment and the eye-tracker are retrieved and the required saliency levels are communicated back to the TPC task environment. Also the current settings can be saved; the participant's name and the IP-address

[2]For more information on Gamemaker, see `http://www.yoyogames.com/gamemaker`.

[3]For more information on Tobii eye-trackers, see `http://www.tobii.com`.

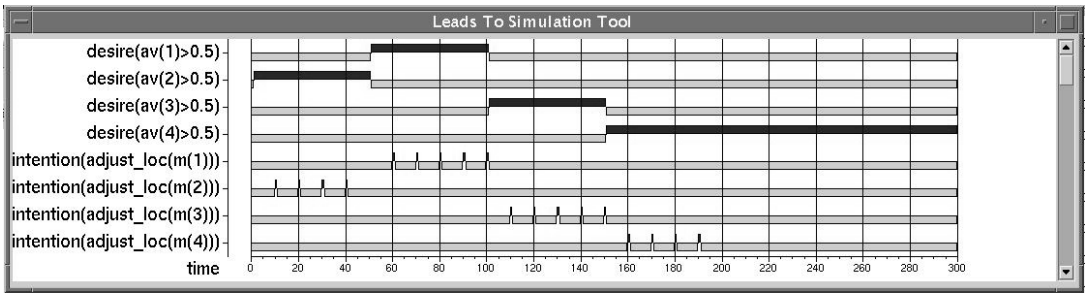


Figure 4. Model-based reasoning process. First it is intended (several times) to adjust a feature value at location 2, then at location 1, then at location 3, and finally at location 4.

where the TPC task environment is running and eye-tracker is connected can be specified here. In the second part, agent model parameters can be set. For this paper we used a type of support where feature manipulation values are to be communicated to the task environment. These values cause the saliency of the different objects on the screen to either increase or decrease, which may result in a shift of the participant's visual attention. As a result, the participant's attention is continuously manipulated in such a way that it is expected that he pays attention to the objects that are considered relevant by the agent. The increase or decrease of the saliency of objects can be done on a continuous or discrete scale, with a binary scale as being discrete. Other types of support can also be set with the support type parameter, such as the estimated threat values of each contact. Furthermore the grid size can be set here. The more fine grained the grid, the more computationally intensive the running of the agent models will be. Time lag is also set here which determines how old (in terms of milliseconds) data is allowed to be in order to be used by the agent models. This is needed because the application is run over a network (though 4 seconds is most likely never reached). The weights together with the decay are the same parameters as also described in Section 3.2. The frequency determines the amount of model loops the agent is allowed to run. The higher the frequency, the more computationally intensive the support agent will be. The third part deals with the frequency of the task environment. This specifies the amount of times the information from the task environments is communicated to the support agent. The fourth part deals with the parameters of the eye-tracker. The frequency specifies the amount of times the gaze location is retrieved. The other options are to visualize the eye-tracker information in real-time or to simulate eye-tracker information by mouse movements instead of gaze behavior.

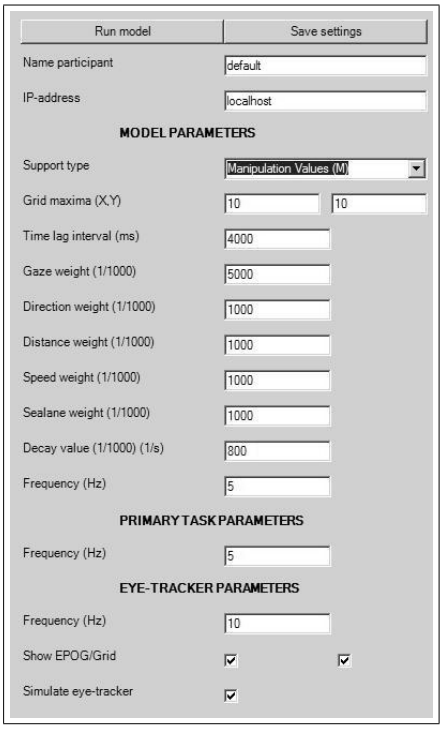


Figure 9. Interface of the attention allocation support system.

5.3 Results

The first results of the agent implemented for this case study are best described by a number of example snapshots of the outcomes of the models used in the agent to estimate (model 1) and manipulate (model 4) attention in three different situations over time (see Figure 10).

On the left side of Figure 10 the darker dots correspond to the agent's estimation of those contacts to which the participant is paying attention. On the right side of the figure, the darker dots correspond to those contacts where attention manipulation is initiated by the system (in this case, by increasing its saliency). On both sides of the figure a cross corresponds to the own ship, a star corresponds to the eye point of gaze, and the x- and y-axes represent the coordinates on the interface of the TPCT. In the pictures to the left, the z-axis represents the estimated amount of attention.

The darker dots on the left side are a result of the exceedance of this estimation of a certain threshold (in this case 0.03). Thus, a peak indicates that it is estimated that the participant has attention for that location.

Furthermore, from top to bottom, the following three situations are displayed in Figure 10:

- After 37 seconds since the beginning of the experiment, the participant is not paying attention to region A at coordinates (7.5,1.5) and no attention manipulation for region A is initiated by the system.
- After 39 seconds, the participant is not paying attention to region A, while the attention should be allocated to region A, and therefore attention manipulation for region A is initiated by the system.
- After 43 seconds, the participant is paying attention to region A, while no attention manipulation for region A is done by the system, because this is not needed anymore.

The output of the attention manipulation system and the resulting reaction in terms of the allocation of the participant's attention in the above three situations, show what one would expect of an accurate system of attention manipulation. As shown in the two pictures at the bottom of Figure 10, in this case the agent indeed succeeds in attracting the attention of the participant: both the gaze (the star in the bottom right picture) and the estimated attention (the peak in the bottom left picture) shift towards the location that has been manipulated.

6 Validation

In order to validate the agent's manipulation model, the results from the case study have been used and tested against results that were obtained in a similar setting without manipulation of attention. The basic idea was to show that the agent's manipulation of attention indeed results in a significant improvement of human performance. Human performance in selecting the five most threatening contacts was compared during two periods of 10 minutes (with and without manipulation, respectively). The type of manipulation was based on determining the saliency of the objects on a binary scale. In this way it was easy (opposed to a continuous scale) to follow the agent's advice. The performance measure took the severity of an error into account. Taking the severity into account is important, because for instance selecting the least threatening contact as a threat is a more severe error than selecting the sixth most threatening contact. This was done by the use of the following penalty function:

$$P_x = \begin{cases} \dfrac{|t_x - \dfrac{t_5 + t_6}{2}|}{\sum_k^{24} p_k} & \text{if } x \text{ incorrectly selected} \\ 0 & \text{otherwise} \end{cases}$$

here p_k is the pre-normalized penalty of contact k, with $p_k = |t_k - \frac{t_5+t_6}{2}|$, and t_x is the calculated threat value of contact x (there are 24 contacts) using the contact's speed, heading, distance to own ship and position in or out a sea-lane. The task performance is then calculated by subtracting the sum of all penalties of the contacts from 1.

After the above alterations, the average human performance over all time points of the condition "support" was compared with the average human performance of the first condition "no support", where "support" ($M+ = 0.8714$, $SD+ = 0.0569$) was found significantly higher (i.e., $p < .05$) than "no support" ($M- = 0.8541, SD- = 0.0667$),

with $t(df = 5632) = 10.46$, $p < .001$. Hence significant improvements were found comparing the first and the second condition. Finally, subjective data based on a questionnaire pointed out that the participant preferred the "support" condition above that of the "no support" condition.

7 VERIFICATION

In addition to this validation, the results of the experiment have been analyzed in more detail by converting them into formally specified traces (i.e., sequences of events over time), and checking relevant properties, expressed as temporal logical expressions, against these traces. To this end, a number of properties were logically formalized in the language TTL (Bosse et al., 2009a). This predicate logical language supports formal specification and analysis of dynamic properties. TTL is built on atoms referring to states of the world, time points and traces, i.e., trajectories of states over time. In addition, dynamic properties are temporal statements that can be formulated with respect to traces based on the state ontology Ont in the following manner. Given a trace γ over state ontology Ont, the state in γ at time point t is denoted by state(γ, t). These states can be related to state properties via the formally defined satisfaction relation denoted by the infix predicate $\models$, comparable to the Holds-predicate in the Situation Calculus: state(γ, t) $\models p$ denotes that state property p holds in trace γ at time t.

Based on these statements, dynamic properties can be formulated in a formal manner in a sorted first-order predicate logic, using quantifiers over time and traces and the usual first-order logical connectives such as $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\forall$, $\exists$. To give a simple example, the property 'there is a time point t in trace tr at which the estimated attention level of space $(1, 2)$ is 0.5' is formalized as follows (see Bosse et al., 2009a):

$\exists t$:TIME state(tr,t) $\models$ belief(has_value(av(1,2), 0.5))

Below, a number of such dynamic properties that are relevant to check the agent's attention manipulation are formalized in TTL, in a similar manner as was done in (Bosse et al., 2007b).[4] To this end, some abbreviations are defined:

discrepancy_at(γ:TRACE, t:TIME, x, y:COORDINATE) $\equiv$
$\exists a, h$:REAL estimated_attention_at(γ,t, x, y, a) &
state(γ,t) $\models$ desire(has_value($av(x, y)$, h)) & $a < h$

This predicate states that at time point t in trace γ, there is a discrepancy at space (x, y). This is the case when the estimated attention at this space is smaller than the desired attention. Next, abbreviation estimated_attention_at is defined:

estimated_attention_at(γ:TRACE,t:TIME,
x, y:COORDINATE, a:REAL) $\equiv$
state(γ,t) $\models$ belief(has_value($av(x, y)$, a))

This takes the estimated attention as calculated by the agent at runtime. This means that this definition can only be used under the assumption that this calculation is correct. Since this is not necessarily the case, a second option is to calculate the estimated attention during the checking process, based on more objective data such as the gaze data and the features of the contacts.

Based on these abbreviations, several relevant properties may be defined. An example of a relevant property is the following[5]:

PP1 Discrepancy leads to Efficient Gaze Movement
If there is a discrepancy at (x, y) and the gaze is currently at (x_2, y_2), then within δ time points the gaze will have moved to another space (x_3, y_3) that is closer to (x, y) (according to the Euclidean distance).

PP1(γ:TRACE, t:TIME, x, y:COORDINATE) $\equiv$
$\forall x_2, y_2$:COORDINATE
discrepancy_at(γ,t,x,y) & state(γ,t) $\models$
gaze_at(x_2,y_2) & $t < LT - \delta \Rightarrow$
$\exists t_2$:TIME $\exists x_3, y_3$:COORDINATE [
$t < t_2 < t + \delta$ & state(γ,t_2) $\models$ gaze_at(x_3,y_3) &
$\sqrt{(x - x_2)^2 + (y - y_2)^2} > \sqrt{(x - x_3)^2 + (y - y_3)^2}$]

[4] Note that the properties introduced in (Bosse et al., 2007b) were used mainly to check whether the attention model (as described in Section 3.2) behaved correctly, whereas the current properties aim to check for successfulness of the attention manipulation model.

[5] Note that this property assumes a given trace γ, a given time point t and a given space (x, y).

In the above property, a reasonable value should be chosen for the delay parameter δ. Ideally, δ equals the sum of 1) the time it takes the agent to adapt the features of the contacts and 2) the person's reaction time.

To enable automated checks, a special software environment for TTL exists, featuring both a Property Editor for building and editing TTL properties and a Checking Tool that enables formal verification of such properties against traces (Bosse et al., 2009a). Using this TTL Checking Tool, properties can be automatically checked against traces generated from any case study. In this paper the properties were checked against the traces from the experiment described in Section 5. When checking such properties, it is useful to know not only if a certain property holds for a specific space at a specific time point in a specific trace, but also how often it holds. This will provide a measure of the successfulness of the system. To check such more statistical properties, TTL offers the possibility to test a property for all time points, and sum the cases that it holds. Via this approach, PP1 was checked against the traces of the experiment with $\delta = 3.0$ sec. These checks pointed out that (under the "support" condition) in 88.4% of the cases that there was a discrepancy, the gaze of the person changed towards the location of the discrepancy. Under the "no support" condition, this was around 80%.

8 Formal Analysis

The results of validation and verification discussed above may ask for a more detailed analysis. In particular, the question may arise of how a difference between 80% without support and 88% with support as reported above should be interpreted. Here a more detailed formal analysis is given that supports the context for interpretation of such percentages. To this end the effect of arbitrary transitions in gaze dynamics is analyzed, in particular those that occur between the time points of monitoring the gaze and adjustment of luminance.

At a given time point, the adjustment of luminance is based on the gaze at that point in time. A question is whether at the time the luminance is actually adjusted, the gaze is still at the same point. When the system is very fast in adjusting the luminance this may be the case. However, it is also possible that even in this very short time the gaze has changed to focus on another location on the screen. Here it is analyzed in how many cases of an arbitrarily changed gaze the luminance adjustment by the system should still be sufficient. The general idea is that this is the case as long as the gaze transition does not increase the distance between gaze location and considered discrepancy location. The area of all locations of the screen for which this is the case is calculated mathematically below; here the worst case is analyzed, the case when the considered discrepancy location is at the corner of the screen. The screen is taken as a square. The function f indicates an under-approximation of the number (measured by the area) of locations with distance at most r to O (see Figure 11, with $r = OQ$). For $r \leq d$ the area within distance r to O is a quarter of a circle: $\pi/4{\cdot}r^2$; so $f(r) = \pi/4{\cdot}r^2$, for $r \leq d$. For $r > d$ an approximation was made. The part of distance to O larger than r is approximated by two triangles as $\triangle PQR$ in Figure 11.

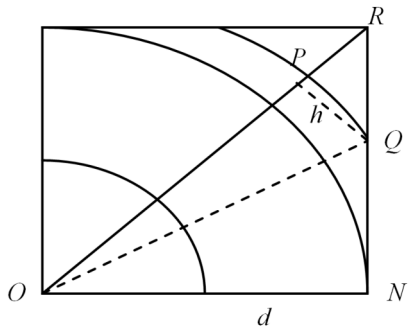


Figure 11. Gaze area approximation.

In Figure 11 the following holds:

$ON = d, \quad QN = \sqrt{r^2 - d^2},$

$QR = RN - QN = d - \sqrt{r^2 - d^2},$

$OR = \sqrt{2} \cdot d, \quad PR = OR - OP = \sqrt{2} \cdot d - r,$

$PQR = \frac{1}{2} PR \cdot h,$ with h the distance of Q to OR,

$h = \frac{1}{2}\sqrt{2} \cdot QR = \frac{1}{2}\sqrt{2} \cdot (d - \sqrt{r^2 - d^2})$

The whole area $-2\triangle PQR$ is

$$d^2 - \frac{1}{2}\sqrt{2} \cdot (d - \sqrt{r^2 - d^2})(\sqrt{2} \cdot d - r)$$

Therefore, for $r > d$, it is taken

$$f(r) = d^2 - \frac{1}{2}\sqrt{2} \cdot (d - \sqrt{r^2 - d^2})(\sqrt{2} \cdot d - r)$$

For $d = 10$ the overall function f divided by the overall area d^2 (thus normalizing it between 0 and 1) is shown in Figure 12. For example, it shows that when $r = \frac{1}{2}d$, then the covered area is around 20% of the overall screen, but when r is a bit larger, for example $r = d$, then at least around 80% is covered. Note that this is a worst case analysis with the location considered in the corner. In less extreme cases the situation can differ. When, for example, the considered location is at the center, then for distance $r = \frac{1}{2}d$, the covered area would be a full circle with radius $\frac{1}{2}d$, so an area of $\pi/4 \cdot d^2$, which is more than 70% of the overall area.

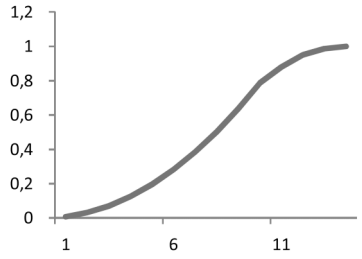


Figure 12. Function of the number of locations within distance r to O, divided by d^2, for $d = 10$.

Moreover, the distance of the considered location where a discrepancy is detected to the actual gaze may not have a uniform probability distribution from 0 to $\sqrt{2} \cdot d$. Indeed, the value 0 may be very improbable, and the larger values may have much higher probabilities. Suppose $\Pr(r)$ denotes the probability (density) that the distance between actual gaze and considered discrepancy location is r, then the expected coverage can be calculated by:

$$\int_0^{\sqrt{2} \cdot d} \Pr(r) \cdot f(r)\, dr$$

For example, if a probability distribution is assumed that is increasing in a modest, linear way from $\Pr(0) = 0$ to $\Pr(\sqrt{2} \cdot d) = 1/d^2$, then for $d = 10$ with $\Pr(r) = r/100$ this becomes approximately (estimated by numerical integration):

$$\int_0^{14} \frac{r \cdot f(r)}{100}\, dr$$

This means that the expected coverage would be 72%. For a bit less modest increase, for example in a quadratic manner for $d = 10$ from $\Pr(0) = 0$ to $\Pr(14) = 0.2$, then the expected coverage is approximately 80% (estimated by numerical integration):

$$\int_0^{14} \frac{r^2 \cdot f(r)}{1000}\, dr = 0.80$$

When it turns out that the gaze is often changing, then a remedy is to base the adjustment of the luminance on a larger distance for r, thus anticipating on the possible future states. The graph for f shows that if r is taken equal to distance d, then a coverage of 80% is achieved.

9 Discussion

An important task in the domain of naval warfare is the Tactical Picture Compilation Task, where persons have to deal with a lot of complex and dynamic information at the same time. To obtain an optimal performance, an intelligent agent can provide aid in such a task. This paper discussed and evaluated an initial version of such a supporting software agent. Within this type of agent an explicitly represented model of human functioning plays an important role, for the case considered here the model of the human's attention.

To obtain a software agent for these purposes, four models were used that are aimed at manipulating a person's attention at a specific location: (1) a dynamical system model for attention, (2) a reasoning model to generate beliefs about attentional states using the attention model for forward simulation, (3) a discrepancy assessment model, and (4) a decision reasoning model, again using the attention model, this time for backward desire propagation. The first two models were adopted from earlier work (Bosse et al., 2009c), and the decision model in (4) from (Bosse et al., 2009b).

After testing the models via simulation experiments, they have been implemented within an ambient agent, in a case study where participants perform a simplified version of the Tactical Picture Compilation Task. Within this case study an experiment was conducted to validate the agent's manipulation. The participants, both in the experiment discussed in this paper as well in earlier pilot studies, reported to be confident that the agent's manipulation indeed is helpful. The results of the validation study with respect to performance improvement have also been positive.

Further investigation has to be done in order to rule out any order effects, which suggests more research with more participants. It is also expected that future improvements of the agent's submodels, based on the gained knowledge from automated verification will also contribute to the improved success of such validation experiments.

A detailed analysis and verification of the behavior of the agent also provided positive results. Traces of the experiment were checked to see whether the agent was able to adapt the features of objects in such a way that they attracted human attention. Results show that when there was a discrepancy between the prescriptive and the descriptive model of attention, the agent indeed was able to attract the human's attention.

Note that the model in this paper assumes mainly a bottom-up influence on attention to a location (i.e., influence of saliency). An existing model that incorporates both bottom-up and top-down aspects of attention is that of (Horrey et al., 2006). Next to the saliency of a location, their model predicts attention taking into account the expectancy of seeing a valuable (important) event at a location and the effort it takes to contribute attention at that location (see also Wickens et al., 2008).

Although top-down influences are not taken into account in the current model, previous research shows that it is possible to extend such models based on a saliency map with top-down features of attention. In (Elazari and Itty, 2010; Navalpakkam and Itti, 2002) a map is proposed that shows the relevancy of a location to the task (task-relevance map) next to the existing saliency map. As our attention model is based on the generic notion of features of a location, it can be easily extended with top-down features as well. In the future, these possibilities will be explored in detail.

Acknowledgments

This research was partly funded by the Royal Netherlands Navy (program number V524).

References

Bosse, T., Both, F., Gerritsen, C., Hoogendoorn, M., and Treur, J. (2008). Model-based reasoning methods within an ambient intelligent agent model. In et al., M. M., editor, *Constructing Ambient Intelligence: AmI-07 Workshops Proceedings*, volume 11 of *LNCS*, pages 352–370. Springer Verlag.

Bosse, T., Jonker, C. M., van der Meij, L., Sharpanskykh, A., and Treur, J. (2009a). Specification and verification of dynamics in agent models. *International Journal of Cooperative Information Systems*, 18:167–193.

Bosse, T., Jonker, C. M., van der Meij, L., and Treur, J. (2007a). LEADSTO: a Language and Environment for Analysis of Dynamics by SimulaTiOn. *International Journal of Artificial Intelligence Tools*, 16(3):435–464.

Bosse, T., Memon, Z., and Treur, J. (2007b). A two-level bdi-agent model for theory of mind and its use in social manipulation. In *Proceedings of the AISB 2007 Workshop on Mindful Environments*, pages 335–342.

Bosse, T., van Lambalgen, R., van Maanen, P.-P., and Treur, J. (2009b). Automated visual attention manipulation. In Paletta, L. and Tsotsos, J., editors, *Proceedings of WAPCV'08, Attention in Cognitive Systems*, volume 5395 of *Lecture Notes in Computer Science*, pages 257–272. Springer-Verlag.

Bosse, T., van Maanen, P.-P., and Treur, J. (2009c). Simulation and formal analysis of visual attention. *Web Intelligence and Agent Systems: An International Journal (WIAS)*, 7(1):89–105.

Chen, L., Xie, X., Fan, X., Ma, W., Zhang, H., and Zhou, H. (2003). A visual attention model for adapting images on small displays. *ACM Multimedia Systems Journal*.

Elazari, L. and Itty, L. (2010). A bayesian model for efficient visual search and recognition. *Vision Research*, 50:1338–1352.

Gore, B., Hooey, B., Wickens, C., and Scott-Nash, S. (2009). A computational implementation of a human attention guiding mechanism in midas v5. In Duffy, V., editor, *Digital Human Modeling, HCII'09*, volume 5620 of *LNCS*, pages 237–246, Berlin Heidelberg. Springer-Verlag.

Heuvelink, A. and Both, F. (2007). Boa: A cognitive tactical picture compilation agent. In *Proceedings of the 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2007)*, page forthcoming. IEEE Computer Society Press.

Horrey, W., Wickens, C., Strauss, R., Kirlik, A., and Stewart, T. (2006). Supporting situation assessment through attention guidance and diagnostic aiding: the benefits and cost of display enhancement on judgment skill. In Kirlik, A., editor, *Human-Technology Interaction*. Oxford University Press, New York.

Itti, L. and Koch, C. (2001). Computational modeling of visual attention. *Nature Reviews Neuroscience*, 2(3):194–203.

Itti, L., Koch, U., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:1254–1259.

Levitt, J. and Lund, J. (1997). Contrast dependence of contextual effects in primate visual cortex. *Nature*, 387:73–76.

Marsella, S., Pynadath, D., and Read, S. (2004). Psychsim: Agent-based modeling of social interaction and influence. In Lovett, M., Schunn, C., Lebiere, C., and Munro, P., editors, *Proceedings of the International Conference on Cognitive Modeling (ICCM 2004)*, pages 243–248.

Memon, Z. and Treur, J. (2008). Cognitive and biological agent models for emotion reading. In Jain, L., Gini, M., Faltings, B., Terano, T., Zhang, C., Cercone, N., and Cao, L., editors, *Proceedings of the 8th IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'08)*, pages 308–313. IEEE Computer Society Press.

Navalpakkam, V. and Itti, L. (2002). A goal oriented attention guidance model. In *Biologically Motivated Computer Vision*, volume 2525 of *Lecture Notes Computer Science*, pages 453–461. Springer.

Nothdurft, H. (2000). Salience from feature contrast: additivity across dimensions. *Vision Research*, 40:1183–1201.

Parkurst, D., Law, K., and Niebur, E. (2002). Modeling the role of salience in the allocation of overt visual attention. *Vision Research*, 42:107–123.

Posner, M. I. (1980). Orienting of attention. *Quarterly Journal of Experimental Psychology*, 32:325.

Sklar, A. and Sarter, N. (1999). Good vibrations: tactile feedback in support of attention allocation and human-automation coordination in event-driven domains. *Human Factors*, 41(4):543–552.

Theeuwes, J. (1994). Endogenous and exogenous control of visual selection. *Perception*, 23:429–440.

Theeuwes, J. (1995). Abrupt luminance change pops out; abrupt color change does not. *Perception & Psychophysics*, 57(5):637–644.

Treisman, A. (1988). Features and objects: The 14th bartlett memorial lecture. *Q. J. Experimental Psychology A*, 40:201–237.

Turatto, M. and Galfano, G. (2000). Color, form and luminance capture attention in visual search. *Vision Research*, 40:1639–1643.

Wickens, C., McCarley, J., Alexander, A., Thomas, L., Ambinder, M., and Zheng, S. (2008). Attention-situation awareness (a-sa) model of pilot error. In Foyle, D. and Hooey, B., editors, *Human Performance Modeling in Aviation*, pages 213–239. Taylor & Francis Group, Florida.

APPENDIX

```
% This matlab code calculates the attention
    values for each time step of a given
    data set:
% v_gaze2 (gaze data)
% v_target4, v_target5a (task environment
    data)
% v_increments (gaze to target data offset
    conversion data)
% a_parameters (model parameters)
% v_model4 (output data)

% constants initialization
xstep = 20; % x-pixel length in grid
ystep = 20; % y-pixel length in grid
c_gridmaxX = 10;
c_gridmaxY = 10;
c_timeinterval = 500; % number of
    milliseconds per time step
a_participantnumber = 1;
i_steps = 2000; % number of time steps
```

```
v_model1 = zeros(i_steps, c_gridmaxX,
    c_gridmaxY); % momentaneous
v_model2 = zeros(i_steps, c_gridmaxX,
    c_gridmaxY); % normalized (1)
v_model3 = zeros(i_steps, c_gridmaxX,
    c_gridmaxY); % temporal
v_model4 = zeros(i_steps, c_gridmaxX,
    c_gridmaxY); % normalized (2)

% if gazeweight = 0 then distance between de
    EPOG and the contact does not
% have any effect, if gazeweight = 'infinity
    ' only the grid coordinates of
% de EPOG will have saliency
gazeweight = 1;

% if decayfactor = 0, then old information
    is not used
% if decayfactor = 1, then new information
    is not used
decayfactor = 0.8;

% initialization model
v_model4(1,:,:) = 1/(c_gridmaxX*c_gridmaxY);
v_model1(1,:,:) = v_model4(1,:,:);
v_model2(1,:,:) = v_model4(1,:,:);
v_model3(1,:,:) = v_model4(1,:,:);

% calculate attention values for each time
    step
for i = 2:i_steps % "2" because "1" is
    already initialized
 summodel = zeros(1,3);
 for x = 1:c_gridmaxX
  for y = 1:c_gridmaxY
   taskfactor = 0; % influence by contacts
       on grid (x,y)
   numberofcontactsonxy = 0;
   if v_increments(i,1,2) > 0 % if there are
       contacts at this timestep
    for k = 1:v_increments(i,1,2) % go
        through all contacts
     if v_target4(v_increments(i,1,3)+k-1,3)
         +1==x && v_target4(v_increments(i
         ,1,3)+k-1,4)+1==y
      % calculate the task factor with the
          previously specified weights per
          participant
      % using a_parameters(for each
          participant, parameternumber)
       average = (a_parameters(
           a_participantnumber, 1)* ...
        v_target5a(v_increments(i,1,3)+k
            -1,2) + ...
        a_parameters(a_participantnumber, 2)
            * ...
        v_target5a(v_increments(i,1,3)+k-1,
            3) + ...
        a_parameters(a_participantnumber, 3)
            * ...
        v_target5a(v_increments(i,1,3)+k-1,
            4)) /...
        sum(a_parameters(a_participantnumber
            , 1:3));
     end
     taskfactor = max(taskfactor, average);
         % use only maximum task factor
     numberofcontactsonxy =
         numberofcontactsonxy + 1;
    end
   end
   if numberofcontactsonxy == 0 % is no
       contacts than use default values
    taskfactor = a_parameters(
        a_participantnumber, 6);
    numberofcontactsonxy = a_parameters(
        a_participantnumber, 7);
   end
   if v_gaze2(i,2) >= 0 % if there is a gaze
    gazefactor = 1/(1 + gazeweight * sqrt( (
        xstep*( x - v_gaze2(i,2) ) ).^2 +
        ...
    ( ystep*( y - v_gaze2(i,3) ) ).^2 ) /
        sqrt(c_primarymaxX.^2 +
        c_primarymaxY.^2));
   else
    gazefactor = 1/sqrt(c_primarymaxX.^2 +
        c_primarymaxY.^2);
   end
   v_model1(i,x,y) = taskfactor*gazefactor;
   summodel = summodel + v_model1(i,x,y);
  end
 end
 % normalize model (1)
 v_model2(i,:,:) = v_model1(i,:,:)/summodel;
end

% merge old with new
for i = 2:i_steps
 summodel = zeros(1,3);

 % calculate real attention values
 for x = 1:c_gridmaxX
  for y = 1:c_gridmaxY
   DECAY = decayfactor.^(c_timeinterval
       /1000);
   OLD = v_model4(i-1,x,y);
   NEW = v_model2(i,x,y);
   v_model3(i,x,y) = OLD*DECAY + NEW*(1-
       DECAY);
   summodel = summodel + v_model3(i,x,y);
  end
 end

 % normalize (2)
 v_model4(i,:,:) = v_model3(i,:,:)/summodel;
end
```

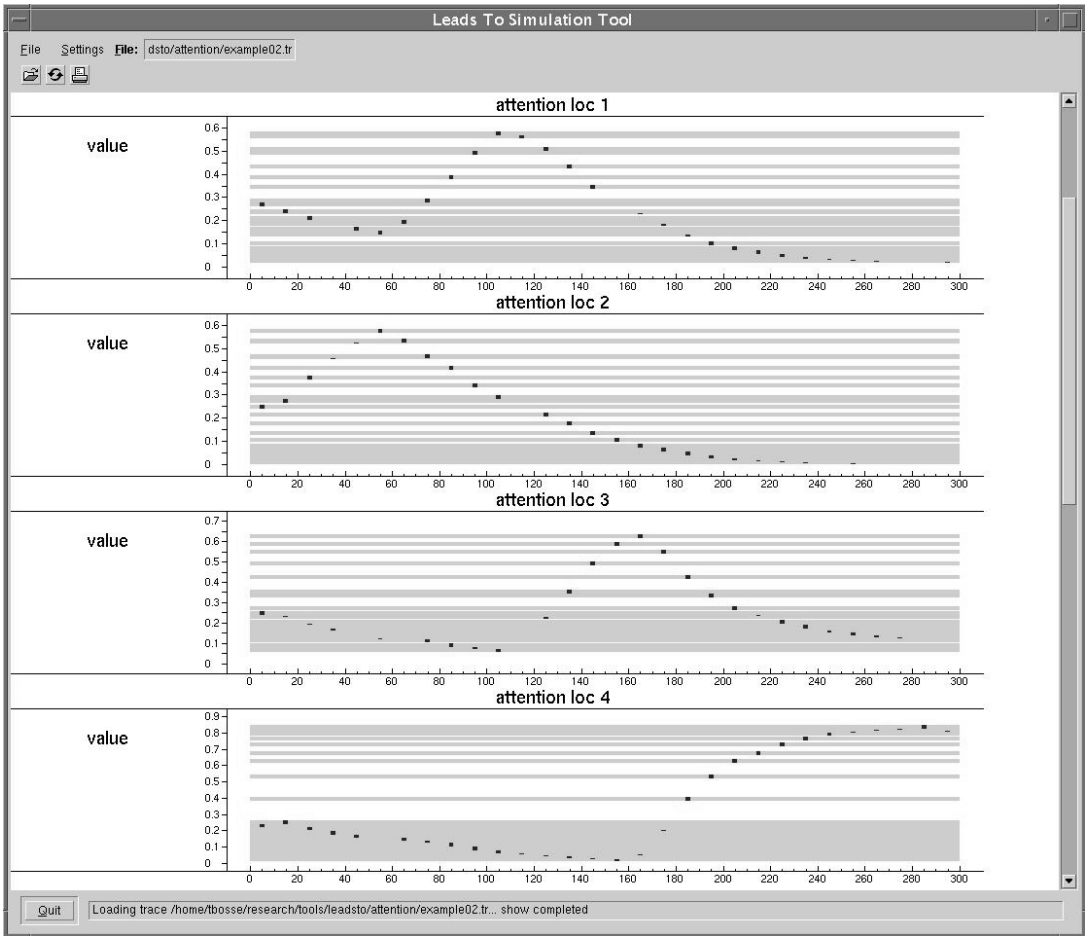


Figure 5. Estimated attention at different locations. Initially the highest attention value is estimated to be at location 2 (with a peak around time point 55), then at location 1, then at location 3, and finally at location 4.

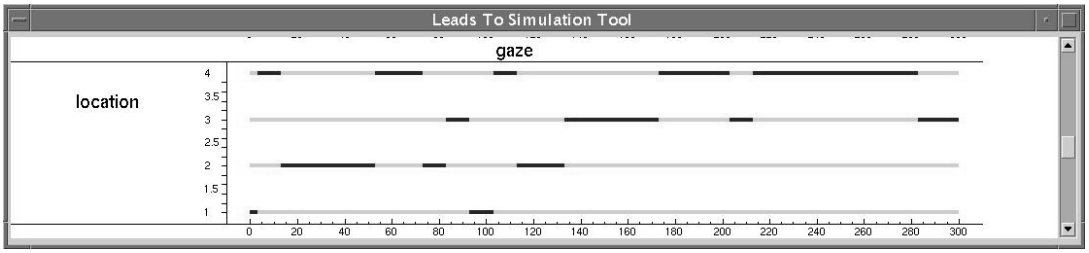


Figure 6. Dynamics of gaze. The vertical axis denotes the location of the gaze, which switches between location 1, 2, 3, and 4.

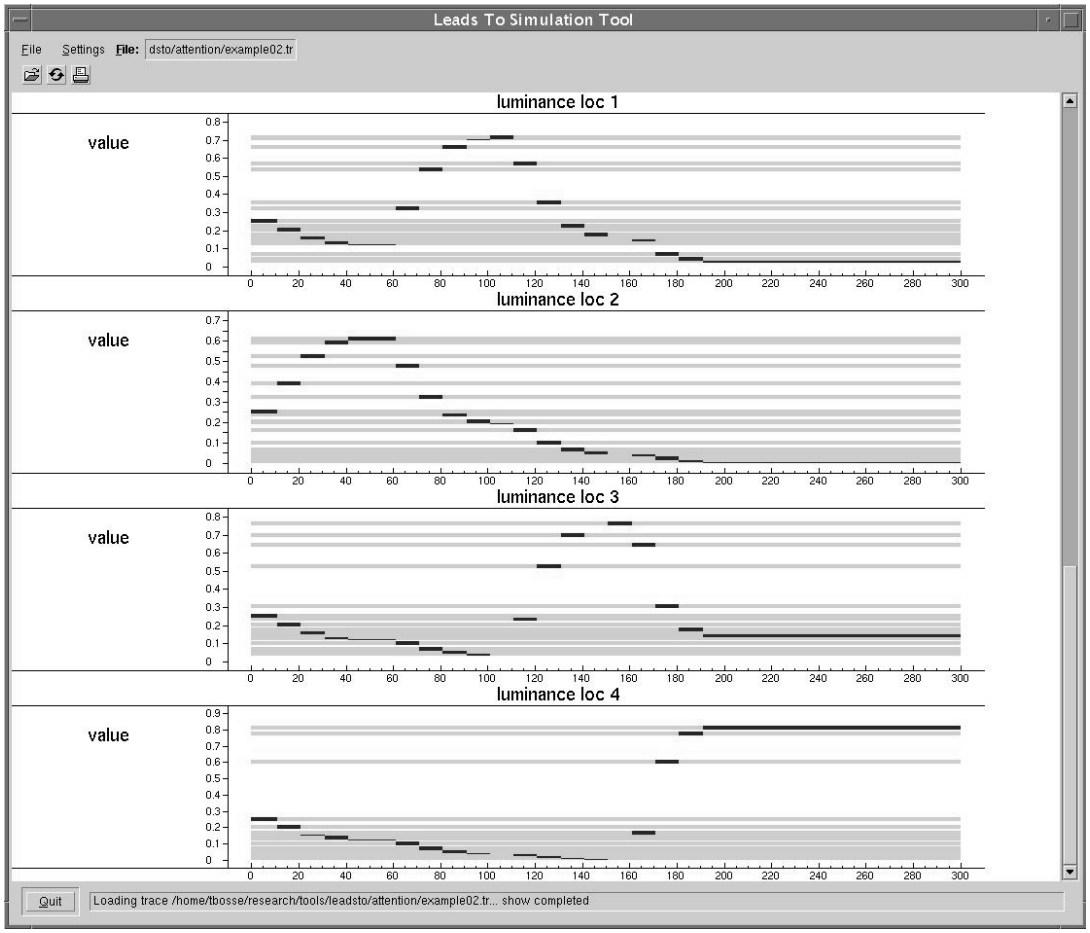


Figure 7. Values of feature 'luminance' at different locations. First the luminance at location 2 is increased, then at location 1, 3, and 4 (note that values are normalized).

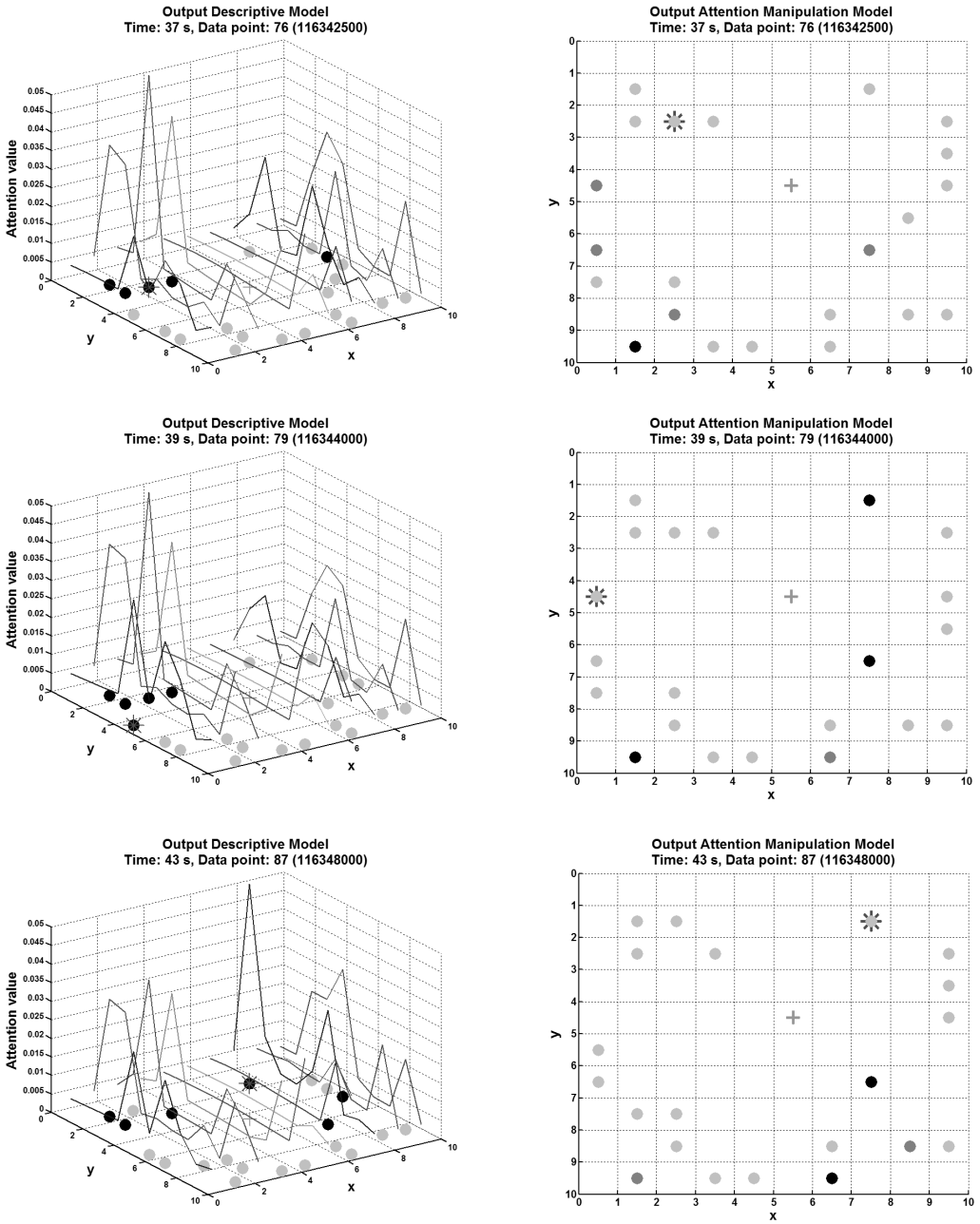


Figure 10. Estimation of the participant's attention division and the agent's reaction.

Chapter 15

Adaptive Attention Allocation Support: Effects of System Conservativeness and Human Competence

Adaptive Attention Allocation Support: Effects of System Conservativeness and Human Competence

Peter-Paul van Maanen*†, Teun Lucassen‡ and Kees van Dongen*

* TNO Human Factors, P.O. Box 23, 3769 ZG Soesterberg, The Netherlands
Email: {peter-paul.vanmaanen, kees.vandongen}@tno.nl
† Department of Artificial Intelligence, Vrije Universiteit Amsterdam
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
‡ Department of Cognitive Psychology and Ergonomics, University of Twente
P.O. Box 215, 7500 AE Enschede, The Netherlands
Email: t.lucassen@gw.utwente.nl

***Abstract*—Naval tactical picture compilation is a task for which allocation of attention to the right information at the right time is crucial. Performance on this task can be improved if a support system assists the human operator. However, there is evidence that benefits of support systems are highly dependent upon the systems' tendency to support. This paper presents a study into the effects of different levels of support conservativeness (i.e., tendency to support) and human competence on performance and on the human's trust in the support system. Three types of support are distinguished: fixed, liberal and conservative support. In fixed support, the system calculates an estimated optimal decision and suggests this to the human. In the liberal and conservative support types, the system estimated the important information in the problem space in order to make a correct decision and directs the human's attention to this information. In liberal support, the system attempts to direct the human's attention using only the assessed task requirements, whereas in conservative support, the this attempt is done provided that it has been estimated that the human is not already paying attention (more conservative). Overall results do not confirm our hypothesis that adaptive conservative support leads to the best performances. Furthermore, especially high-competent humans showed more trust in a system when delivered support was adapted to their specific needs.**

1 Introduction

In the domain of naval warfare, information volumes for navigation, system monitoring and tactical tasks will increase while the complexity of the internal and external environment also increases (Grootjen et al., 2006). For tactical picture compilation tasks also the dynamics in behavior and ambiguity of threat is expected to increase. Furthermore, the trend of reduced manning is expected to continue as a result of economic pressures and humans will be responsible for more and more demanding tasks. Although attention can be divided between tasks, problems with attention allocation and task performance are expected since attentional resources are limited (Wickens, 1984; Kahneman, 1973). Experience, training and better interfaces can lift these limitations, but only to a certain level. Even with experienced users, attentional problems are still likely to occur (Pavel et al., 2003).

Automation can assist humans by directing attention to critical events (Wickens and McCarley, 2007). It can heuristically identify and prioritize objects of interest by highlighting high priority objects and dimming low priority objects. This helps humans to focus on the right subset of objects and thereby effectively reduces the number of objects that must be monitored. The downside is that this form of cuing can impede the detection of important objects that are mistakenly left unhighlighted when the automation is imperfect or when the situation is uncertain (St. John et al., 2005). In these situations problems with inappropriate trust in the automation, resulting in over-reliance on the automation, are expected.

In naval warfare, problems with inappropriate trust and reliance on decision support may lead to disastrous consequences. Operators may trust the decision support too much (over-trust) by, for in-

stance, unjustly accepting advice on whether a contact is (not) threatening. Unfortunately, 'threat' is an ill-defined and complex function of many attributes and correctly assessing the level of threat requires years of training and experience (St. John et al., 2005). There are several reasons why reliable advice about threat level from decision support systems is unlikely to be achieved. Assessing threat level can become too complex when attempting to account for all possible variables. Assessments can be ambiguous because important data may be unknown or unknowable or because there is no consensus between experts on criteria. An operator might for instance monitor only those contacts indicated as a threat, missing those that were incorrectly assessed as low threats. Aids that mistakenly overrate the threat level of a contact may induce operators to treat it more aggressively than necessary. Naval operators may also come to distrust advice when unreliability of the decision aid is recognized. In this case they will rely on their own assessment when they trust the aid's threat assessment less than their own.

In this paper the effect is investigated of different types of adaptive decision support with respect to system conservativeness (high low) and human competence (good and poor) in terms of task performance, trust, understandability and responsibility. The paper is composed of the following sections. In Section 2 the related theoretical background and related work is discussed. In Section 3 the proposed support types are described and several hypotheses are given and motivated about these support types. Then, in Section 4 a experiment is described in which the support types are evaluated. The results of this experiment are given in Section 5 and the paper ends with a discussion and conclusions in Section 6.

2 Background

2.1 Unreliable Automation

According to Parasuraman et al. (2000, p. 293) it is often not possible to automate decision making completely due to the fact that reliability is not guaranteed. During all information processing phases, but especially during the decision making phase (as compared to the information acquisition, information analysis and action implementation phases (e.g., Wickens, 1992)) all kinds of factors play a role, of which many are difficult to measure (Dekker and Woods, 2002). Wrong decisions made by automation that can have severe consequences are not accepted by users (Miller and Funk, 2001, p. 1).

Earlier research by Crocoll and Coury (1990) has shown that a high level of automation during the information analysis and decision making phase of air defense tasks leads to a reduction of the time needed to identify aircraft, provided the automation is reliable.

Skitka et al. (1999, p. 1002) have shown that when reliability of the automation of the above mentioned phases is 100%, the number of errors decreases. They also find that when the reliability is low, the number of wrong decisions increases up to above the level of the situation where no automation is applied.

Moray et al. (2000, pp. 52–53) found that the monitoring of unreliable automation requires such an amount of visual and mental resources of a human, that it causes a decrease of performance in important parallel tasks.

Unreliability has different effects on performance dependent on the particular information processing phase one is in. Research of Crocoll and Coury (1990) shows that unreliability in information provided by automation has a larger negative impact when automating the decision making phase compared to the information acquisition phase. Comparable results were found in a different context by Rovira et al. (2002) and in yet another context by Sarter and Schroeder (2001). This implies that one should not automate the decision phase too much when to some extent it can still be considered unreliable.

Another effect of unreliable automation is that in many situations humans do not play proactive roles. Humans are poor performers with respect to monitoring machines (Schutte, 1999). Parasuraman et al. (1993) used the term complacency to point to the effect that general performance decreases due to the fact that performance on the monitoring decreases when the human plays a less active role in the task.

To conclude, in a multiple task environment resources for information processing can be freed up by automating (parts of) tasks. The then available

resources can be used to execute other tasks, provided that the automation is reliable enough in order not to cause any excessive monitoring efforts by the users. And when it comes to monitoring automation, another effect, called complacency, gives rise to yet another negative effect of over-automation. The proper way of dealing with such issues could be to adapt to the current need for automated support taking the above negative effects into account.

2.2 Adaptive Automation

Hilburn et al. (1997, p. 84) define adaptive automation as follows: "adaptive automation refers to a system capable of dynamic, workload-triggered reallocations of task responsibility between human and machine". There are multiple reasons for applying adaptive automation. How well people perform a certain task is affected by the allocation of their attention. People that are more experienced will be better at dividing attention between different sources of information. Research on the effects of playing video games has shown that visual attention abilities often improve with training. Experienced players of video games required less attentional resources for a given target (Green and Bavelier, 2003). In the case of a tactical picture compilation task, experts will be able to track more contacts. Experts will also be able to determine more quickly whether a contact is a possible threat. As opposed to poor performers, good performers will apply the rules correctly. Adaptive automation can help by assisting (the less well performing) humans in their allocation of attention through estimating their current allocation of attention and intervene when the human should reallocate his attention.

Another important factor affecting task performance is cognitive task load. This tends to be higher when a task is more difficult or when multiple tasks require attention in parallel. In an experiment of Scallen and Hancock (2001), a critical event in one task was used to trigger reallocations of task responsibility between human and machine in another task. Scallen and Hancock (2001) showed that their adaptive automation caused a decrease of cognitive task load when it was needed.

Kaber et al. (2005) mention that little has been done to study the effect of automating cognitive tasks. In an experiment they studied the effect of adaptive automation in the information acquisition and action implementation phases compared to that in the information analysis and decision making phases. Opposed to the earlier described research, in Kaber et al. (2005)'s studies no *indicators* of task performance degradation are used as critical events to trigger adaptivity of automation, but rather performance *itself*.

Kaber et al. (2005) also showed that it is important for the the user of the support system to know that the applied automation indeed improves (their) performance. Lee and See (2004), for instance, showed that appropriate trust in automation decreases when automation does not properly show its effect to performance. Kaber et al. (2005) conclude that future research on adaptive automation should focus on the prevention of unwanted cognitive load due to ineffective automation and mis-calibration of trust in automation.

Parasuraman and Riley (1997) propose to use adaptive automation to prevent 'complacency' and to increase the changes of detecting errors. By shifting task responsibility between humans and machines, humans will be more involved in tasks, which causes more errors to be detected and therefore the performance with respect to monitoring automation will increase.

3 Attention Allocation Support

3.1 Generic Support Model

One way of implementing adaptive automation is to use computational cognitive models of attention as a basis for triggering change in automation. A cognitive model of attention is a model which estimates a human's focus of attention at each moment in time for a given task (see e.g., Bosse et al., 2009b,a). Together with a normative model, which estimates where attention should be optimally allocated for that same moment in time and task, a decision support system can aid the user in distributing limited attentional resources when there is a large difference between the two. Bosse et al. (2009b,a) showed, for instance, that in this way it is possible to support humans in their allocation of attention.

The support evaluated in this study has three variants, namely the *fixed*, *liberal* and *conservative* support type. The fixed support is defined as support

that advices a human user what decision to make, without taking into account whether it is needed to support the human at that moment. The outcome of the task is shown to the human who can then decide whether to comply with the advice or to rely on his own judgment. As stated earlier, a potential risk of fixed support is inappropriate reliance. The fixed support system always gives its advice. So the easiest way to perform the task is to follow the advice as given by the system, which can lead to problems with complacency. This means that if the fixed system occasionally gives incorrect advice, it is more likely to be (incorrectly) taken over by the human, compared to an adaptive support system.

The alternative for the fixed support system is to direct the attention of the human to areas that are estimated to need human attention, instead of suggesting a specific decision. This way, the human is supported during an earlier stage of information processing, namely information acquisition, and hence leaving information interpretation and decision making to the human. The result is that the human can no longer completely rely on the support with respect to deciding what to do. Errors in the support are thus likely to be less influential on the decisions of the human. Wrong advice of the support system is also expected to be detected more easily by humans, because the advice is checked more thoroughly due to the fact that it needs to be processed more before a decision is made. This basic idea of bringing the human back 'in the loop' is also the underlying property of the last two support variants.

The liberal and conservative support type are different with respect to system conservativeness. *System conservativeness* is defined as the inverse tendency of the system to provide support to the human. It can be varied through adaptation to the behavior of the human. Examples of this behavior are mouse clicks, reaction times to events, and point of gaze. The models used for liberal adaptive support will use less behavioral data than those for conservative adaptive support. For instance, in the context of the tactical picture compilation task described by Bosse et al. (2009b,a), liberal can be defined as support that adapts only to the current selection of threatening contacts. In this case, it is estimated (by a mathematical model) whether support is needed through adaptation to the clicking behavior of a human operator. For the conservative support, next to adaptation to the clicking behavior, an estimation of the current human attention allocation is also incorporated. Overall this means that the liberal support is less adapted to the user than the conservative support.

3.2 Hypotheses

For the above mentioned three support types, the effects on 1) task performance, 2) trust, 3) understandability and 4) responsibility are studied, which are discussed in sections 3.2.1, 3.2.2, 3.2.3 and 3.2.4, respectively. In sections 3.2.5 and 3.2.6 overall effects of system conservativeness and human competence are discussed. The above mentioned discussions lead to a total of 6 hypotheses.

3.2.1 Task performance: When support is fixed, humans are expected to be more prone to over- and under-rely on the automation, whereas with adaptive support less problems with inappropriate reliance or complacency are expected. This can be explained by the fact that fixed support allows humans to rely entirely on the support (i.e., just take over the computer's advice). Adaptivity can stimulate the human's involvement in the task by automatically applying support, only where and when needed. It is expected that higher levels of such adaptivity to the human also results in higher task performance. This basically boils down to the following hypothesis:

Hypothesis 1. *The proposed adaptive support results in higher task performance than fixed (non-adaptive) support.*

3.2.2 Trust: Another important factor is trust in the support: Do participants trust adaptive decision support more than fixed decision support? Errors will inevitably occur in the support. However, these errors are likely to be much more salient in when applying fixed support. This boils down to the following hypothesis:

Hypothesis 2. *The proposed adaptive support results in more trust in the support system compared to fixed support.*

3.2.3 Understandability: A potential problem in adaptive support is understandability. Adaptive support systems are likely to be more complex than

fixed support systems (or in any case, no support). This leads to the following hypothesis:

Hypothesis 3. *The proposed adaptive support results in a poorer understanding of the support compared to fixed support.*

3.2.4 Responsibility: Since we expect the human to be more involved in the task when applying the proposed adaptive support, we also expect that the responsibility for a good result as felt by the human is higher.

Hypothesis 4. *The proposed adaptive support results in a greater feeling of responsibility for the eventual outcome compared to fixed support.*

3.2.5 System Conservativeness: As has been mentioned, conservativeness can be varied through adaptation to the behavior of the human. This adaptivity can come in various degrees: A more conservative adaptive support system depends to a higher degree on the behavior of the human. When the system is uncertain about information, conservative support will withhold information longer than liberal support. This is expected to result in a stronger effect with respect to task performance, trust and understandability. This boils down to the following hypothesis:

Hypothesis 5. *The claimed effects in Hypotheses 1 to 4 are stronger for conservative adaptive support than for liberal adaptive support.*

3.2.6 Human Competence: We expect that, since adaptive support takes actions of the human into account, the task performance of the human (with or without support) contributes to the performance of the support. When the actions of the human are in line with the task model of the support, the support will be more appropriate. The hypothesis:

Hypothesis 6. *The claimed effects in Hypotheses 1 to 4 are stronger for good performers than for poor performers.*

4 Method

4.1 Participants

Forty college students (17 male, 23 female) with an average age of 23.9 years ($SD = 2.6$) participated in the experiment as paid volunteers.

4.2 Apparatus

Participants had to perform a (simplified) naval tactical picture compilation task as performed in naval warfare. The goal of this task is to build up awareness of possible threats surrounding the own ship (contacts). A screenshot of the task environment is shown in Figure 1.

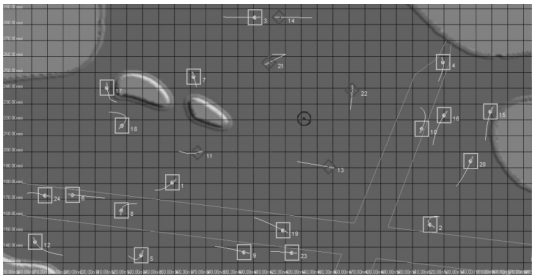


Figure 1. Screenshot of the task environment.

Participants had to mark the five most threatening contacts by clicking on them. To determine if a contact is a possible threat the following criteria were used: speed, heading, distance to own ship and position in or out a sea-lane. All criteria were equally important. The five contacts scoring highest on these criteria had to be selected as most threatening. The behavior of the contacts was such that the threat varied over time. For instance, a contact could get out of a sea lane, speedup, or change its heading toward the own ship. Contacts that were mistakenly identified as threats (false alarms) or contacts that were mistakenly not identified as threats (misses) resulted in a lower task performance.

All participants were exposed to the same task complexity. The complexity was determined by the ambiguity and the dynamics of the behavior of the contacts. Concerning ambiguity, small differences in the threat level of contacts made it more difficult to identify the five most threatening contacts. Dynamics were determined by the varying number of threat level changes of contacts over time. Changes in threat levels were such that the number of times that each contact needed to be re-evaluated was high.

The task including the different developed adaptive support conditions (see Section 4.4) was implemented using the game implementation software

Gamemaker.[1]

4.3 Design

A 4 (system conservativeness) $\times$ 2 (human competence) mixed design was used. System conservativeness was a within-subjects independent variable and the order was balanced between the participants. Human competence was a between-subjects quasi-independent variable.

4.4 Independent Variables

Two independent variables were used: system conservativeness and human competence.

System Conservativeness: There were four levels of conservativeness for the used support system: no support, fixed support, liberal adaptive support and conservative adaptive support.

In the no support (NS) condition participants performed the tactical picture compilation task without any form of support.

The fixed support (FS) used a task model to determine the five most threatening contacts. Threat levels were determined by measuring and weighing the previously mentioned criteria for each contact. Position in or out a sea-lane was a binary variable, the other criteria were calculated relative to the maximum possible value. The threat level of each contact is the sum of the measured values for all criteria.

The reliability of the task model was manipulated by adding errors to the five most threatening contacts. This manipulation was done in order to simulate an imperfect task model. If the task model was perfect there was not use of comparing FS with the other support types, since it would always be a good decision to follow the advice of FS, resulting in a maximum task performance. Reliability manipulation was done by swapping the threat level of a number of contacts in the top 5 with contacts in places 5 to 10 in the task model. The number of swapped threat levels varied between 1 and 4, such that the reliability varied over a 10 minute trial. Three different orders of reliability were used to eliminate learning effects and interaction effects with the scenarios used. The three orders were respectively 20%-80%-50%-80%-20%, 80%-50%-80%-20%-20%, and 20%-20%-80%-50%-80%, in which the percentages represent the number of contacts with incorrect threat levels in the top 5 of most threatening contacts. The duration of each error level was 2 minutes. Note that the average reliability was always the same.

The five most threatening contacts according to the task model were highlighted in white in the interface. The other contacts were displayed in a darker shade of gray.

The liberal adaptive support (LAS) highlighted two types of contacts. Firstly, the two contacts with the lowest threat levels which are already selected by the user were highlighted. This was done because they are candidates for deselection and should therefore receive attention. Secondly, the three contacts with the highest threat levels which are not selected by the user were highlighted. This was done because they are candidates for selection and should receive attention for this reason. This principle is shown in Figure 2.

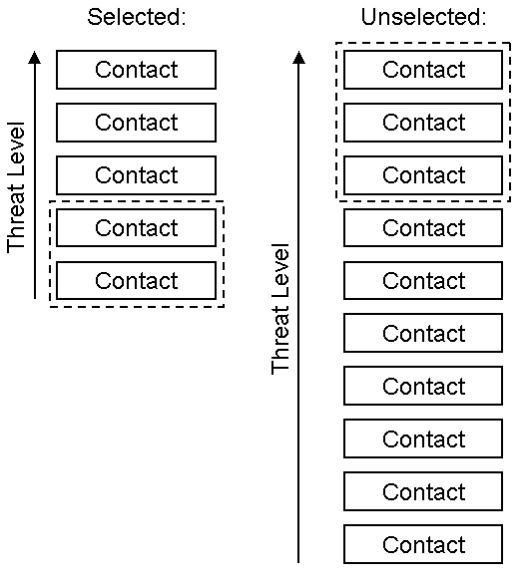


Figure 2. Liberal adaptive support model. Contacts are ranked according to threat level within the selected and unselected contacts. Contacts within the dotted rectangles are highlighted by the support.

The same errors as in the fixed support condition

[1]For more information on Gamemaker, see http://www.yoyogames.com/gamemaker.

were added in this support type. Note that the highlighted contacts are independent of the correctness of the selection of the user. Incorrectly selected contacts will always be highlighted since they will have the lowest threat level of all selected contacts. The same holds for incorrectly unselected contacts, these will have the highest threat levels of all unselected contacts.

The conservative adaptive support (CAS) basically highlighted the same contacts as the liberal support (as indicated with the dotted rectangles in Figure 2, but now only when the user paid little attention to these contacts. Attention values for all contacts were calculated using the cognitive model of attention proposed by Bosse et al. (2009b,a). This cognitive model of attention was implemented in C# and was running over the network on another computer to estimate the current attention division of the participant in (near) real-time in order to determine when support was needed: Only when the attention value for the contacts highlighted in the liberal support condition was below a predetermined threshold, they were highlighted in the conservative model. This means that a always a maximum of 5 contacts was highlighted. When the attention of the user was estimated to be allocated to the 5 contacts normally highlighted in the LAS condition, actually none of these contacts were highlighted. Again, the same errors as in the fixed support condition were added to the task model.

Human Competence: After the experiment, a median split on the task performance in the NS condition was performed to distinguish a good and poor human competence group.

4.5 Dependent Variables

Four dependent variables were measured: task performance, trust, understandability and responsibility.

Task Performance: The task performance on the tactical picture compilation task was determined by measuring the accuracy of the identification of the five most threatening contacts during the task. The task performance measure took the severity of errors into account. This is important, because for instance identifying the least threatening contact as a threat is a more severe error than mistakingly identifying the sixth most threatening contact as a threat. The accuracy was measured using the penalty function P_x:

$$P_x = \begin{cases} \dfrac{|t_x - \dfrac{t_5 + t_6}{2}|}{\sum_{k}^{24} p_k} & \text{if } x \text{ incorrectly selected} \\ 0 & \text{otherwise} \end{cases}$$

where p_k is the pre-normalized penalty of contact k, with $p_k = |t_k - \frac{t_5+t_6}{2}|$, and t_x is the calculated threat value of contact x (there are 24 contacts) using the contact's speed, heading, distance to own ship and position in or out a sea-lane. The task performance is then calculated by subtracting the sum of all penalties of the contacts from 1.

Trust: After each trial participants estimated the reliability of the support system on a scale between 0 and 100% correct. Since trust is for an important part determined by perceptions of reliability (Lee and See, 2004; Gao and Lee, 2006; Dzindolet et al., 2001), this was considered as a good measure of trust.

Understandability: Participants rated after each trial the degree to which they thought the decision making process of that of the support system was understandable on a 7-point Likert scale between -3 (not understandable) and 3 (understandable).

Responsibility: Participants rated after each trial the degree to which they felt responsible for the outcome of the task on a 7-point Likert scale between -3 (not responsible) and 3 (responsible).

4.6 Procedure

The criteria that had to be used for the tactical picture compilation task were thoroughly explained before the experiment using various examples. After this, the participants were given a test on paper to check whether they were able to correctly apply the criteria. All answers were explained afterwards. After the test, the knowledge of the participants was sufficient to start the task. In order to get used to the task and its interface, the participants performed a practice trial of 5 minutes in which they had to perform the task under supervision. After this, the actual experiment began. Each experiment

contained four trials of 10 minutes, each with a different system conservativeness condition. Before each trial, the participant was instructed on how to use the support type (when available) in the upcoming trial. After each trial a questionnaire on this particular condition had to be filled in concerning trust, understandability and responsibility. The task performance was automatically retrieved and stored on a hard disk drive. Between each trial, there was a 5 minute break. Afterwards the participants were debriefed to double check whether there were any problems during the experiments.

4.7 Statistical Analysis

The computer that was used for running the cognitive models in (near) real-time was also used for gathering all necessary data for later statistical analysis using the Statistics Toolbox of Matlab (for data pre-processing and descriptive statistics) and Statistica (for inferential statistics).[2] The exact statistical methods used are mentioned in the next section.

5 Results

Two out of the 40 retrieved data sets have been removed due to unintended errors during the experiment. All participants passed the test on paper and were therefore expected to be able to correctly apply the classification criteria.

5.1 Task Performance

Lilliefors tests have shown that task performance in the NS, FS, LAS and CAS conditions were all normally distributed (i.e., null hypothesis that they are normally distributed could not be rejected).

To check whether the design of the fixed support system was a fair competitor for the adaptive variants, at least it should hold that FS results in higher performances than NS. This was indeed the case: Participants in condition FS ($M = 89.24$, $SD = 2.20$) performed significantly better compared to NS ($M = 87.62$, $SD = 3.27$), $t(74) = 2.53$, $p < .001$.

[2]Since the description of the C#, Gamemaker software and Matlab scripts used for this experiment is out of the scope of this paper, those further interested in this are referred to `http://www.few.vu.nl/~pp/attention`.

Figure 3 shows the main effect of system conservativeness on task performance. A repeated measures analysis of variance (ANOVA) showed a significant main effect ($F(2, 72) = 27.40$, $p < .001$).

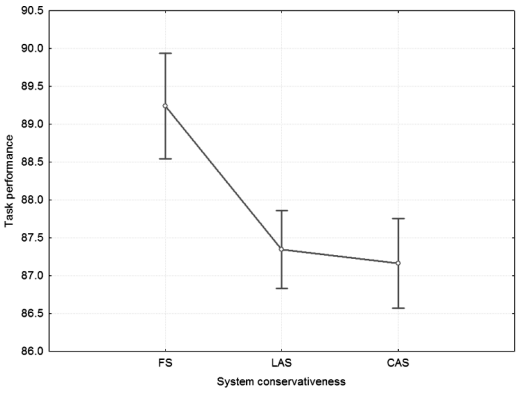


Figure 3. Main effect of system conservativeness for task performance.

A post-hoc Bonferroni test showed that there is a significant difference between conditions FS and LAS ($p < .001$), FS and CAS ($p < .001$), but not between LAS and CAS ($p = 1$). Hence participants performed worse in the LAS ($M = 87.34$, $SD = 1.73$) and CAS condition ($M = 87.16$, $SD = 1.87$) than in the FS condition ($M = 89.24$, $SD = 2.20$). Hypotheses 1 and 5 (for task performance) are therefore not accepted.

Figure 4 shows the possible interaction effect between system conservativeness and human competence on task performance. No interaction effect was found ($F(2, 72) = 0.22$, $p = .80$). Hence Hypothesis 6 (for task performance) is not accepted.

5.2 Trust

Figure 5 shows the main effect of system conservativeness on trust. A repeated measures analysis of variance (ANOVA) did not show a significant main effect ($F(2, 72) = 0.47$, $p = .63$). Hypotheses 2 and 5 (for trust) are therefore not accepted.

Figure 6 shows the interaction between system conservativeness and human competence for trust. A significant interaction effect was found ($F(2, 72) = 3.17$, $p = .048$).

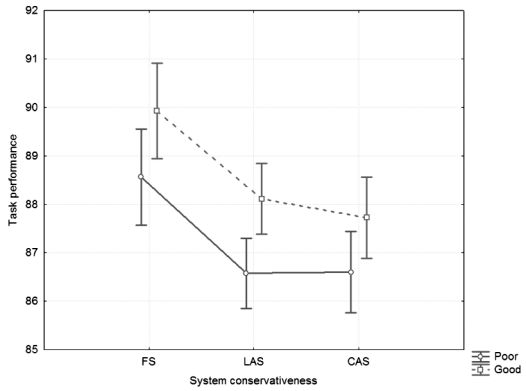


Figure 4. Interaction effect between system conservativeness and human competence for task performance.

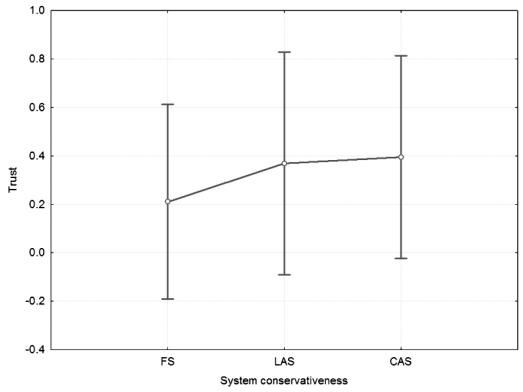


Figure 5. Main effect of system conservativeness for trust.

A post-hoc Bonferroni test showed that there is a significant difference in trust between the good ($M = 1.11$, $SD = 0.29$) and poor ($M = -0.32$, $SD = 0.29$) competence group in the CAS condition ($p = .02$), but not in the LAS ($p = 1$) and FS ($p = 1$) condition. Hence the claimed effect in Hypothesis 2 is stronger for good performers than for poor performers in the case of CAS. For CAS, Hypothesis 6 (for trust) is therefore accepted, but not for LAS.

5.3 Understandability

Figure 7 shows the main effect of system conservativeness on understandability. A repeated measures analysis of variance (ANOVA) did not show a significant main effect ($F(2, 72) = 0.42, p = .66$). Hypotheses 3 and 5 (for understandability) are therefore not accepted.

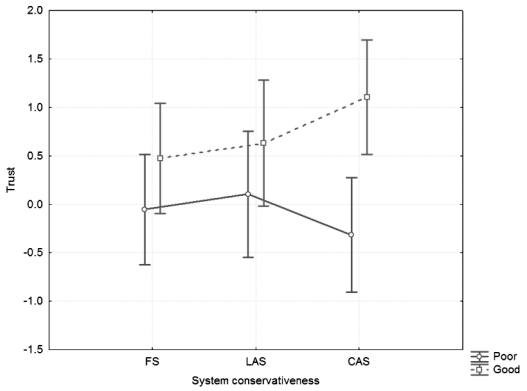


Figure 6. Interaction effect between system conservativeness and human competence for trust.

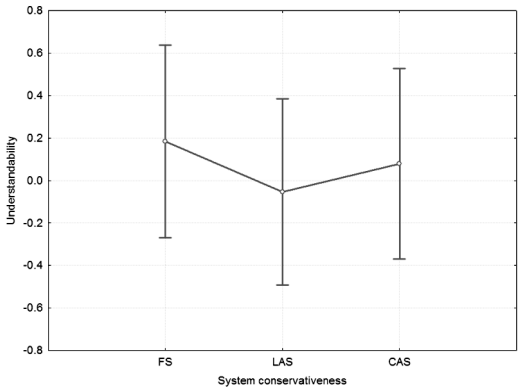


Figure 7. Main effect of system conservativeness for understandability.

Figure 8 shows the interaction between system conservativeness and human competence for understandability. No significant interaction effect was found ($F(2, 72) = 0.92$, $p = .40$). Hypothesis 6 (for understandability) is therefore not accepted.

5.4 Responsibility

Figure 9 shows the main effect of system conservativeness on responsibility. A repeated measures

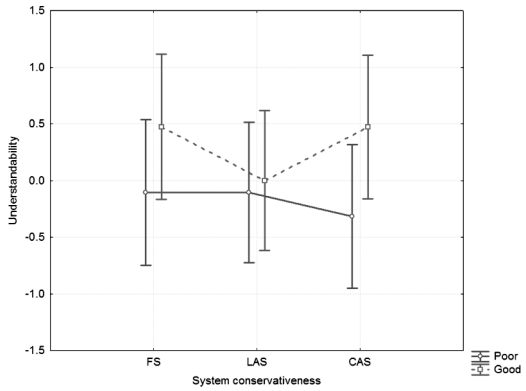


Figure 8. Interaction effect between system conservativeness and human competence for understandability.

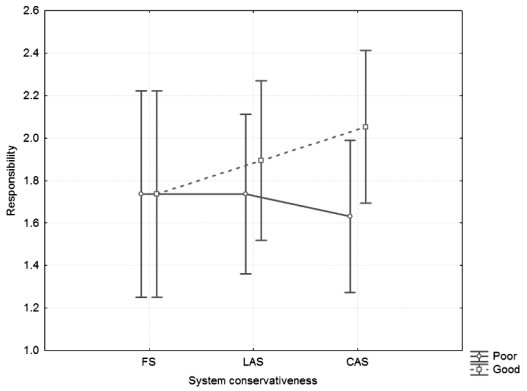


Figure 10. Interaction effect between system conservativeness and human competence for responsibility.

analysis of variance (ANOVA) did not show a significant main effect ($F(2, 72) = 0.37, p = .69$). Hypotheses 4 and 5 (for responsibility) are therefore not accepted.

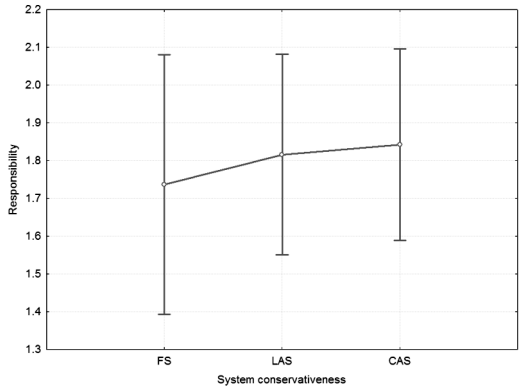


Figure 9. Main effect of system conservativeness for responsibility.

Figure 10 shows the interaction between system conservativeness and human competence for responsibility. No significant interaction effect was found ($F(2, 72) = 1.39, p = .26$). Hypothesis 6 (for responsibility) is therefore not accepted.

6 Conclusion and Discussion

In this study we investigated the benefits of adaptive attention allocation support over fixed (non-adaptive) support in a tactical picture compilation task. We expected task performance using adaptive support to be higher than in fixed support. However, this first hypothesis was not accepted. Trust in adaptive and fixed support did not differ significantly, also rejecting the second hypothesis. In contrary to our third hypothesis, our participants did not report to have a poorer understanding of the more complicated adaptive support than of the fixed support. Also the fourth hypothesis, stating that the feeling of responsibility would be higher in the adaptive condition, could not be accepted based on the results in this study.

The influence of system conservativeness and human competence was also investigated on the first four hypotheses on task performance, trust, understandability and responsibility. The results did not show a significant effect of system conservativeness on any of these variables, so the fifth hypothesis could not be accepted.

For human competence, the effect was significant for trust, but only in the conservative adaptive support condition. This means that well performing participants had more trust in conservative adaptive support than poorly performing participants. This confirms the sixth hypothesis (for trust), but only for the conservative condition. The sixth hypothesis could not be accepted for task performance, understandability, and responsibility. The increase of trust in the conservative adaptive support for

good performers can be explained by the effect that good performers are more likely to understand the task and the effect support systems have on task performance. Lee and See (2004), for instance, show that the use of automation decreases when the effect of automation to performance is not properly perceived.

There are several possible explanations for why an increase of task performance was not found in our experiment. Our implementation of adaptive support aimed at reducing inappropriate reliance on fixed support. However, this comes with a cost in the form of added complexity. Although participants did report a clear understanding of how both adaptive support systems worked, it is still possible that the disadvantage of added complexity is larger than the advantages of such a system. Working with complex (support) systems can raise the cognitive load on the human, leaving less capacity to focus on the actual monitoring of contacts. In this case, this resulted in a significantly higher task performance in the fixed support condition than in both adaptive support conditions. Future design of adaptive support systems should aim at keeping the system as simple as possible, though preserving the expected advantages of adaptivity.

Another possible explanation is the fact that naive participants without prior training or experience in tactical picture compilation were used. In spite of a considerable pre-experiment training, inherent to this naivety are inter-personal learning differences. These learning differences possibly lead to higher deviations in task performance compared to when experts would have been used. Also more training could be fruitful, but this comes with the risk of fatigue at the end of the experiment.

For the adaptive support investigated in this study, it was not possible for the human to simply follow suggestions of the support system. This was because, instead of suggesting a possible answer to a problem, only areas of interest were indicated by the system. This meant that, in any case, the proposed adaptive support must have eliminated inappropriate reliance on the support. The found results in this study are not a reason for rejecting this principle and therefore more research on adaptive attention allocation support is suggested, focusing on the *requirements* in which such a system can help to gain task performance.

Acknowledgments

This research has partly been supported by the programme Cognitive Modeling" (V524), funded by the Dutch Ministry of Defense. The authors would also like to thank Karel van den Bosch, Tibor Bosse, Anja Langefeld and Jan-Willem Streefkerk for their helpful comments.

References

Bosse, T., van Lambalgen, R., van Maanen, P.-P., and Treur, J. (2009a). Attention manipulation for naval tactical picture compilation. In *Proceedings of the 2009 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'09)*.

Bosse, T., van Lambalgen, R., van Maanen, P.-P., and Treur, J. (2009b). Automated visual attention manipulation. In Paletta, L. and Tsotsos, J., editors, *Proceedings of WAPCV'08, Attention in Cognitive Systems*, volume 5395 of *Lecture Notes in Computer Science*, pages 257–272. Springer-Verlag.

Crocoll, W. M. and Coury, B. G. (1990). Status or recommendation: Selecting the type of information for decision aiding. In *Proceedings of the Annual Meeting of the Human Factors Society*, pages 1524–1528, Santa Monica, CA.

Dekker, S. W. A. and Woods, D. D. (2002). MABA-MABA or Abracadabra? Progress in human-automation co-ordination. *Cognition, Technology, and Work*, 4:240–244.

Dzindolet, M. T., Beck, H. P., Pierce, L. G., and Dawe, L. A. (2001). A framework of automation use. Technical Report ARL-TR-2412, Army Research Laboratory, Aberdeen Proving Ground, MD.

Gao, J. and Lee, J. D. (2006). Extending decision field theory to model operator's reliance on automation in supervisory control situations. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 36(5):943–959.

Green, C. S. and Bavelier, D. (2003). Action video game modifies visual selective attention. *Nature*, 423:534–537.

Grootjen, M., Neerincx, M. A., and van Weert, J. C. M. (2006). Task-based interpretation of operator state information for adaptive support.

In *Foundations of Augmented Cognition: Strategic Analysis*, LNCS, pages 236–242. Springer, Arlington, Virginia, 2 edition.

Hilburn, B., Jorna, P., Byrne, E., and Parasuraman, R. (1997). The effect of adaptive air traffic control (atc) decision aiding on controller mental workload. *Human-Automation Interaction: Research and Practice*, pages 84–91.

Kaber, D. B., Wright, M. C., Prinzel, L. J., and Clamann, M. P. (2005). Adaptive automation of humanmachine system information-processing functions. *Human Factors*, 47:730–741.

Kahneman, D. (1973). *Attention and effort*. Prentice Hall, Englewoods Cliffs, NJ.

Lee, J. D. and See, K. A. (2004). Trust in automation: Designing for appropriate reliance. *Human Factors*, 46(1):50–80.

Miller, C. and Funk, H. (2001). Issues in user acceptance and human/machine performance: Lessons learned from fielding intelligent, adaptive information systems. In *Proceedings of SIGGRAPH 2001*.

Moray, N., Inagaki, T., and Itoh, M. (2000). Adaptive automation, trust, and self-confidence in fault management of time-critical tasks. *Journal of Experimental Psychology: Applied*, 6(1):44–58.

Parasuraman, R., Molloy, R., and Singh, I. L. (1993). Performance consequences of automation-induced "Complacency". *International Journal of Aviation Psychology*, 3:1–23.

Parasuraman, R. and Riley, V. A. (1997). Humans and automation: Use, misuse, disuse, abuse. *Human Factors*, 39:230–253.

Parasuraman, R., Sheridan, T. B., and Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics*, 30:286–297.

Pavel, M., Wang, G., Li, K., and Li., K. (2003). Augmented cognition: Allocation of attention. In *Proceedings of 36th Hawaii International Conference on System Sciences*, pages 286–300. IEEE Computer Society.

Rovira, E., McGarry, K., and Parasuraman, R. (2002). Effects of unreliable automation on decision making in command and control. In *Proceedings of the Annual Meeting of the Human Factors Society*.

Sarter, N. and Schroeder, B. K. (2001). Supporting decision-making and action selection under time pressure and uncertainty: The case of in-flight icing. *Human Factors*, 43(4):573–583.

Scallen, S. F. and Hancock, P. A. (2001). Implementing adaptive function allocation. *International Journal of Aviation Psychology*, 11:197–221.

Schutte, P. (1999). Complemation: an alternative to automation. *Journal of Information Technology Impact 1*, pages 113–118.

Skitka, L. J., Mosier, K. L., and Burdick, M. (1999). Does automation bias decision-making? *International Journal of Human-Computer Studies*, 51(5):991–1006.

St. John, M., Smallman, H. S., Manes, D. I., Feher, B. A., and Morrison, J. G. (2005). Heuristic automation for decluttering tactical displays. *Human Factors*, 47:509–525.

Wickens, C. D. (1984). Processing resources in attention. In Parasuraman, R. and Davies, D. R., editors, *Variaties of attention*, pages 63–101, Orland, FL. Academic Press.

Wickens, C. D. (1992). *Engineering psychology and human performance*. HarperCollins, New York, 2nd edition.

Wickens, C. D. and McCarley, J. S. (2007). *Applied attention theory*. CRC Press, Boca Raton, FL.

Part IV

Research Overview and General Discussion

Chapter 16

Research Overview and General Discussion

As was stated in the introduction of this dissertation, the objective of the reported research was to investigate means for integrating knowledge of the human factor in human-computer cooperation into the reasoning capabilities of support systems. This is done to reduce the amount of problems caused by insufficient mutual understanding of the capabilities and limitations of humans and of support systems. The goal to increase reasoning capabilities of support systems was reached by incorporating executable cognitive models, which describe human cognition as accurately as possible, including its limitations, into these systems. Subsequently, these cognitive models are used to detect occurrences of limitations. Limitations were detected by the comparison of the output of two types of cognitive models: one that describes the current cognitive state (i.e., a *descriptive cognitive model*) and one that prescribes the desired cognitive state (i.e., a *prescriptive cognitive model*). Such limitation detections were then used as triggers for adaptation of the support to the human need for assistance, ideally resulting in an increase, or prevention of a decrease, of human-computer team performance. The specific adaptive support explored in this thesis focused on *adaptive autonomy* and *decision support*. The specific cognitive models explored in this thesis focused on *trust* and *attention*.

This chapter is composed of two sections: Section 1 is an overview of the research reported in this thesis. This is done by going through all methodological phases introduced in Chapter 1 and the chapters in which those phases were used. In this section also the most important conclusions and possible future research is discussed per chapter. Section 2 is a short general discussion of this thesis.

1 Research Overview

The research methodology outlined in Chapter 1 was used for pursuing the above stated research objective. In Table 1 it is shown which phases of the methodology were used in which chapters. The different phases are on the vertical axis. The different chapters of the

		Part II (Trust)						Part III (Attention)						
		3	4	5	6	7	8	9	10	11	12	13	14	15
a	Determination of domain and related Human Factors issues	+	+	−	−	−	−	+	−	+	−	−	−	−
b	Development of informal cognitive models	+	−	+	−	−	−	−	+	−	+	+	+	−
c	Psychological experimentation	+*	−	+	−	−	−	−	−	−	+	−	−	+
d	Formalization of cognitive models	+	−	−	+	+	−	−	+	−	+**	+	+	−
e	Verification of cognitive models	−	−	−	−	+	−	−	+	−	−	−	+	−
f	Validation and tuning of cognitive models	+*	−	−	−	+	−	−	−	+	+	+	−	−
g	Development of adaptive support system	+*	+*	−	+	−	+	+*	−	+	−	−	+	+
h	Verification of adaptive support system	−	−	−	+	−	−	−	−	−	−	−	+	−
i	Evaluation of adaptive support system	+*	+*	−	+	−	+	−	−	+	−	−	+	+

* only the plans or preliminary results from this methodological phase were reported in the chapter
** methodological phase was used in the study, but was not reported in the chapter

Table 1: Overview of the methodological phases described in the different chapters of this thesis.

thesis (except the chapters in Part I, IV and V) are on the horizontal axis. A "+" indicates that a certain methodological phase was used in the study described in a corresponding chapter and a "−" that it was not used. Note that the reason for the table not being completely filled with pluses is not that the corresponding phases were impossible to be used, but rather that the focus in the particular chapters was not on those phases.

Since the application of the research methodology used in this thesis was yet to be explored, the exact content in Table 1 could not have been determined before the studies in the different chapters were performed. Now that these studies *have* been performed, the application of the method can be further evaluated and described. This is why the application of the method is further discussed in this chapter rather than in the introduction, where it was first introduced. Below the above mentioned description and evaluation is given, together with an overview of the main conclusions and future research.

(II) *Trust*
In Part II, two chapters used methodological phase a, two chapters phase b, two chapters phase c, three chapters phase d, one chapter phase e, two chapters phase f, four chapters phase g, one chapter phase h and four chapters phase i. Two experimental environments have been developed (a pattern learning and a classification task environment in Chapter 3 and 7, respectively) for which four types of support have been developed and evaluated (two in Chapter 6 and two in Chapter 8), us-

ing four different variants of cognitive models of trust (one in Chapter 3, one in Chapter 6 and two in Chapter 7).

(3) **Towards Task Allocation Decision Support by means of Cognitive Modeling of Trust**

Research methodology: First, in Chapter 3 the Human Factors issues related to trust and automation reliance were explored and discussed (a). Then informal descriptive and prescriptive cognitive models of attention were described (b) and formalized (d). In this chapter, descriptive trust was formalized as estimated trust of an agent i in another agent j concerning the execution of a certain action α. Prescriptive trust was formalized as the estimated trust of an 'infallible agent $*$'. A first description was given of an adaptive support system that was able to reallocate tasks dynamically using cognitive models of trust (g*)[1]. A design was described of an experiment with an implemented task environment (a *pattern learning task*, where people had to predict the next number out of 1, 2 and 3, given a certain pattern of past correct answers) to gain more insight into the Human Factors issues (c*) and to validate the above mentioned cognitive model (f*) and to evaluate the above mentioned support system (i*).

Main conclusions: The results were of the exploratory kind and no definite conclusions could be drawn.

Future research: The described experimental environment should be used for further research on extensions of the proposed cognitive model of trust and dynamical task allocation, such as on indirect acquisition of knowledge (e.g., reputation, gossip), analogical judgments, allocation engagement costs (e.g., waiting, cooperation, and overhead costs), allocation implementation errors, level of autonomy, the allocation decision inhibitory bound, quantity and seriality of tasks and time pressure. It also is suggested that future research on cognitive modeling of trust should aim at support in the four stages of information processing (Parasuraman et al., 2000): the acquisition of information relevant for trust, its integration to trust concepts, task allocation decision making based on trust concepts and the implementation of the allocation decision.

(4) **Closed-Loop Adaptive Decision Support Based on Automated Trust Assessment**

Research methodology: In Chapter 4 the implemented task environment and Human Factors issues related to trust and automation reliance in Chapter 3 were further developed and explored, respectively (a). First descriptions of different support systems (g*) and some preliminary evaluation results were presented and discussed (i*). The support systems were variants of the in

[1]The marks "*" and "**" are related to the same footnotes as in Table 1.

Chapter 3 first described support system and augmented human cognition with respect to the human's cognitive task to calibrate trust and make reliance decisions. The goal of augmented cognition is to extend the human's cognitive performance via the development and use of computational technology, such as the envisioned support systems. The support systems had different autonomy settings: minimal autonomy, maximal autonomy and adaptive autonomy support. The *minimal autonomy support* assisted the human by giving advice related to trust and reliance decision making (called Operator Reliance Decision Making (Operator-RDM)), the *maximal autonomy support* took over reliance decision making (called the RDM Model (RDMM)) and *adaptive autonomy support* could dynamically decide between the two former support types (called Meta-RDMM).

Main conclusions: First results showed that human reliance decision making was not perfect and could be augmented by computational decision making. Maximal autonomy support (RDMM) turned out to be the best with respect to the human-computer team performance as compared to the other support types (Operator-RDM and Meta-RDMM).

Future research: It has been recommended that future research should focus on the investigation of how human-machine cooperation can be augmented in more complex and more realistic situations. It should be further explored whether models of trust and reliance can be practically used to adjust the level of autonomy of adaptive systems and in what domains this kind of support has an impact on the effectiveness of task performance, and how the magnitude of the impact depends on the task's and the domain's characteristics.

(5) **Reliance on Advice of Decision Aids: Order of Advice and Causes of Under-Reliance**

Research methodology: In Chapter 5 different established cognitive psychological theories and (informal) models about trust and reliance behavior were discussed (b) and several hypotheses related to the order of advice and the causes of mis-calibration of trust were tested in psychological experiments (c), using two further developed versions of the experimental environment introduced in Chapter 3 (the pattern learning task). The two versions were different with respect to the order of the advice given (i.e., either the advice of the human first or that of the support system).

Main conclusions: Several main conclusions could be drawn based on the results from this chapter. First of all, the results showed that a 'self bias' (i.e., an *a priori* tendency to trust oneself more than another, and the support system more specifically) can be observed. The results also showed that people disagree more with a support system when they express their decision before rather than after receiving advice from the support system. The results furthermore showed that this is only the case when decision makers trust themselves more than the support system. No self bias was found when trust in

the support system exceeded trust in oneself. It was therefore argued that in existing frameworks of automation use, the notion of automation bias needs to be complemented with that of the self bias. Whether self biases lead to desirable outcomes or not, depends on whether perceptions of reliability of one's own performance and that of the support system are appropriate. When people wrongly think they perform better than the support system, self reliance can result in undesirable outcomes. The results showed that decision makers rely less on the support system than what would be expected based on relative trust in performance reliability (difference between trust in oneself and the other) alone: The participants did not rely more often on the support system, although they perceived it to be 30% more reliable. The results further suggested that decision makers rely less on conflicting advice because they perceive the advisor's reasoning to be cognitively less available and understandable than their own reasoning. The results showed that people who felt more responsible for the task outcome relied more on conflicting advice than people who feel less responsible. And finally, perceived reliability of both oneself and the support system was underestimated when feedback about performance was provided and it was found that negative experiences have a greater influence than positive experiences.

Future research: It was argued that appropriate reliance on support systems is not guaranteed when only focusing on optimizing the reliability of these systems. Several other things should also be done during the design phase: One of the important recommendations was that it might help when future support systems are able to give feedback about performance of humans and their support systems, but correct for the bias that negative information is given more weight. This feedback can improve the calibration of trust in oneself and the support system and therefore stimulate appropriate reliance and trust calibration. Secondly, by providing advice after, rather than before, more knowledge is brought to the task. Such a design would not be focused on reducing workload by automation, but focused on human-computer collaboration with the goal of increasing accuracy and resilience. Also, it was recommended to make people feel accountable for the outcomes of the human-computer team. That is, hold people responsible for the quality of outcome of the human-computer team. Finally, it was argued that one should control for the attribution of errors. For instance by making sources of error transparent or by making operators aware of their biases in attribution. The idea was that providing information regarding why the automation might be mistaken reduces inappropriate distrust (Dzindolet et al., 2003).

(6) **Aiding Human Reliance Decision Making Using Computational Models of Trust**

Research methodology: In Chapter 6 a more elaborate variant of the prescriptive cognitive model of trust introduced in Chapter 3 was formalized and

tailored to the in Chapter 4 described support types (d). The second and third support types from Chapter 4 (maximal (RDMM) and adaptive autonomy support (Meta-RDMM)) were further developed and implemented (g). The dynamics of the support system were simulated for the purpose of verification (h) and validation (i). The general goal of the developed support system was to improve performance of human-computer teams either by taking over reliance decision making using trust models calibrated by the support system itself (RDMM), or by deciding adaptively when the human or the system makes the reliance decision (Meta-RDMM).

Main conclusions: Overall, the results showed that indeed calibration of trust and intervention by the computer can lead to an increase of human-computer team performance. The participants may have performed worse than (Meta-)RDMM because of limited attentional and memory resources and biases in weighing successes and failures of both themselves and the support system. The results showed a substantial amount of occurrences (above chance) in which humans made better reliance decisions than the support system. It was suggested that this could mean that reliance decision making completely done by the support system does not result in an optimal performance. This could be explained by the asymmetric availability of the underlying reasons for possible decreases of performance (i.e., human compared to support system performance) and the possibility of applying these reasons to the current situation. Meta-RDMM tried to take advantage of this, but without result.

Future research: Further extension of the model and exploration of the above mentioned principle behind Meta-RDMM was said to belong to the possible future research. Since the support systems have been simulated, one possibility to indeed find a significant effect of Meta-RDMM is to apply the support with a 'human in the loop', which might imply lower human performance degradation due to less problems with complacency as compared to RDMM when a large part of the task is taken over by the system.

(7) **Validation and Verification of Agent Models for Trust: Independent Compared to Relative Trust**

Research methodology: In Chapter 7 two variants of descriptive cognitive models of trust (the independent and relative trust model) were formalized (d), verified (e), validated (f) and compared to each other. A different experimental environment was used (a *classification task environment*). The *independent trust model* was inspired on, but different from, the formalized model in Chapter 6: the model now could estimate trust in three trustees and was used as a descriptive instead of a prescriptive model (the human was assumed to be similar as what the system would think an infallible agent would do). The difference between the independent and the *relative trust model* was that for the relative trust model the estimated human's trust in a certain trustee also depended on estimations for trustees that are considered competitors of

that trustee (an additional modeled psychological phenomenon). The used experimental environment was contextually more rich than the environment introduced in Chapter 3 (the pattern learning task environment) and required cooperation between two humans and two computers. The task was now comparable to tasks in specific areas related to target identification based on video footage (but therefore also most probably less comparable to other specific areas not related to that).

Main conclusions: The results showed that both an independent as well as a relative trust model can predict reliance behavior with a high accuracy (72% and 80%, respectively). Furthermore, the results also showed that underlying assumptions of the trust models were found in the data of the participants (s.a. the underlying assumption that if on average more positive experiences of a trustee are identified, the advice of that trustee is also more often relied upon).

Future research: It was argued that future research should aim at exploring or extending other parameter adaptation methods for the purpose of real-time adaptation. Furthermore, it was mentioned that future research will focus on the development of support systems that monitor and balance the functional state of the human for optimal performance for all kinds of tasks in different domains, such as the military, aviation or air traffic control domain.

(8) **Effects of Reliance Support on Team Performance by Advising and Adaptive Autonomy**

Research methodology: In Chapter 8 two types of support systems (graphical and adaptive autonomy support) based on the second type of cognitive model of trust from Chapter 7 (for descriptive trust) and a variant of the model from Chapter 6 (for prescriptive trust) were developed (g), evaluated and compared to no support (i). The idea behind the *graphical support* was that trust calibration and reliance decision making was supported by an advice from the support system, whereas *adaptive autonomy support* could take over reliance decision making, using its own trust models.

Main conclusions: The results showed that team performance in the different support conditions was somewhat higher compared to no support. However, these differences were not significant. A significant increased effect was found for participants that performed less well. The results also showed significantly less satisfaction when applying adaptive autonomy compared to advising through the graphical support.

Future research: Future efforts should aim at investigating what precisely can go wrong when humans make reliance decisions, why this is such a difficult task for humans and how to provide leverage for exactly that. It was stressed that possible future variants of reliance decision support should aim at making the usage of the support less intrusive. Future improvement of the cognitive models of trust should also improve support systems based on those models. Research should also aim at investigating new efforts for taking away

reasons for possible human intolerance for increased machine autonomy in making (important) decisions. Finally, it is mentioned that further research should investigate whether it is of benefit for adaptive team support to also include other psychological and environmental influences, such as analogical judgments and allocation engagement costs (as was already mentioned in Chapter 3).

(III) *Attention*

In Part III, two chapters used methodological phase a, four chapters phase b, two chapters phase c, four chapters phase d, two chapter phase e, three chapters phase f, four chapters phase g, one chapter phase h and three chapters phase i. Three experimental tasks have been developed (an air traffic control, naval tactical picture compilation and shooting game task environment in Chapter 9 (first two) and 13 (last one)) for which five types of support have been developed and evaluated (one in Chapter 11, one in Chapter 14 and three in Chapter 15), using seven different variants of cognitive models of attention (one in Chapter 10, three in Chapter 12, two in Chapter 13 and one in Chapter 14).

(9) **Augmented Meta-Cognition Addressing Dynamic Allocation of Tasks Requiring Visual Attention**

Research methodology: In Chapter 9 the different Human Factors issues related to the dynamic allocation of attention were explored and discussed (a). Furthermore, two preliminary descriptions of applications of attention model-based adaptive support were given (g^*). These descriptions were applied to two introduced experimental environments (i.e., an *air traffic control task* and a *tactical picture compilation task*). The envisioned support systems were able to dynamically allocate tasks based on the comparison between the estimation of the human's current (descriptive) and desired (prescriptive) attentional state.

Main conclusions: The results were of the exploratory kind and no definite conclusions could be drawn.

Future research: In this chapter it was stated that the Augmented Cognition Society defined 'Augmented Cognition' as "an emerging field of science that seeks to extend a user's abilities via computational technologies, which are explicitly designed to address bottlenecks, limitations, and biases in cognition and to improve decision making capabilities." It was furthermore mentioned that Augmented Cognition research is a wide area, that is applicable to various types of cognitive processes. As the area develops further, it may be useful to differentiate the field a bit more, for example, by distinguishing augmented cognition focusing on *task content* versus augmented cognition focusing on *task coordination*. As the latter is considered a form of meta-cognition (coordination of cognitive tasks), this suggests augmented meta-cognition as an interesting sub-area of future augmented cognition support systems. Especially in tasks involving multiple stimuli that require fast re-

sponses, this concept is expected to provide a substantial gain in effectiveness of system support systems.

(10) **Simulation and Formal Analysis of Visual Attention**

Research methodology: In Chapter 10 an informal (descriptive) model of attention and of different attentional states was described (b) and formalized (d). The model was described as being part of the design of an agent-based system (g^*) that is able to monitor a human in the execution of the first task introduced in Chapter 9 (the air traffic control task). The output of the model was simulated using eye-tracker data from humans executing a the task and different expected properties of the model were verified against the simulation data (e).

Main conclusions: The model was specifically tailored to domain-dependent properties retrieved from a task environment; nevertheless it was expected that the method presented in the chapter remains generic enough to be easily applied to other domains and task environments. Furthermore, although the work reported focused on a practical application context, as a main contribution, also a formal analysis was given for attentional states and processes. Using this analysis, it has been proven that it is possible to identify different attentional states and processes, which can be used as additional triggers for adaptivity in support systems.

Future research: The study focused on formal analysis. Although in this formal analysis also empirical data were involved, a more systematic validation of the models put forward in the intended application context should be a next step. Future studies should further focus on the use of estimates of different attentional states and processes for dynamically allocating tasks as a means for assisting humans, as this kind of adaptive human-computer team support may turn out to be fruitful. Open questions are related to modeling both endogenous and exogenous triggers and their relation in one model. Finally, the attention model may be improved and refined by incorporating more attributes within saliency maps, for example based on literature (e.g., Itti and Koch, 2001; Itti et al., 1998; Sun, 2003).

(11) **Design and Validation of HABTA: Human Attention-Based Task Allocator**

Research methodology: In Chapter 11 it was further explored what kind of applications are needed given the found Human Factors issues related to over- and under-allocation of attention (a). Moreover, two experiments were described in which the cognitive model of attention from Chapter 10 was validated (f) and in which a developed adaptive attention allocation support system (g) was evaluated (i). The used experimental environment was based on the second environment introduced in Chapter 9. The support system was

described as an adaptive cooperative agent assisting humans by managing its own and the human's attention. The component involved in the agent's attention management was called *HABTA: The Human-Attention-Based Task Allocator*.

Main conclusions: The results were of the exploratory kind and no definite conclusions could be drawn.

Future research: The results of both experiments presented in this chapter could be seen as a 'proof of concept' and large-scale experiments with multiple participants still needed to be performed. Furthermore, an idea was to compare the HABTA-component to the attention management capabilities of humans, where it is the human who allocates attention of himself or the support agent to different subtasks. In this way the effectiveness of HABTA-based support could be studied more convincingly. It is also stressed that future research should also focus on the development and validation of *prescriptive* cognitive models and not only on descriptive models: what would the system do when it were in the shoes of the human? Finally, in general, agent-components have more value when they can be easily adjusted for other applications. It was therefore said that it would be interesting to see whether HABTA-based support could be applied in other domains as well.

(12) **Effects of Task Performance and Task Complexity on the Validity of Computational Models of Attention**

Research methodology: In Chapter 12 three variants (the gaze-based, task-based and the combined model) of the attention model from Chapter 10 were described informally in relation to task complexity and performance (b), based on which several psychological hypotheses (c) as well as hypotheses related to the validity of models (f) were formed and experimentally tested. Before these models could be tested they had to be formalized, but this was not reported in this chapter because this was not the focus of the chapter (d^{**}). The *gaze-based model* only used the human's gaze data as input for the estimation of the human's attentional state, the *task-based model* only used information from the task and the *combined model* was a combination of the former two. The models were applied to the second task introduced in Chapter 9 (the tactical picture compilation task).

Main conclusions: The results showed that overall, the estimation of the combined model was better than that of the other two models. Contrary to what was expected, the performance of the models was not different for good and bad performers and was not different for simple and complex scenarios. The difference in complexity and performance might not have been strong enough.

Future research: It was mentioned that further research is needed to determine if improvement of the combined model is possible with additional features, such as the interpretation of mouse behavior or the inclusion of a more

elaborate task model. This could be done using a similar validation methodology as was presented in this chapter. To enhance the performance of the models, optimal parameter values need to be determined. Furthermore, since the Area Under the Curve (AUC) performance measure is decision criterion-independent, it was mentioned that it needs to be determined whether liberal or conservative criterion settings are more effective for the estimation or prediction of human attentional states or whether this criterion should be determined dynamically.

(13) **Personalization of Computational Models of Attention by Simulated Annealing Parameter Tuning**

Research methodology: In Chapter 13 the cognitive model of attention from Chapter 10 was personalized. First, this personalization was described and motivated (b), after which the personalization process was formalized (d). The personalized models were tuned and validated using data from humans executing a *shooting game task* and compared to non-personalized models (f). Similar as in Chapter 7 about trust, the usage of other environments for experimentation was expected to lead to better understanding of the scalability and the further possibilities of using cognitive models in adaptive support systems. The personalization of the cognitive model of attention was done by tuning specific model parameters (using simulated annealing (SA)) that were related to certain human personality characteristics.

Main conclusions: Results showed that the attention model with personalization results in a more accurate estimation of an individual's attention as compared to the model without personalization.

Future research: The validation was subjective in the sense that a participant's own estimation was measured by asking to which objects they had directed their attention before certain freezes during the task execution. Future research should also focus on using objective measures. A possible way of measuring objective attention is by looking at mouse clicks at a location. It should be noted that SA is a probabilistic procedure and therefore is suboptimal, specifically as the necessary computing capacity becomes relatively smaller compared to the problem space. In the future, personalization of attention models can be extended. In the personalized model presented in this chapter, parameters were tuned that are known to differ per individual. However, in future research personalization can be done by using collected data on personality to improve the attention model. Furthermore, in the current personalized model, parameters like the attention threshold and the total amount of attention were static. These could be coupled to a individual's functional state (e.g., experienced pressure, exhaustion), making the model fit for each individual, but also in different conditions (high or low workload). Such adjustments were expected to result in again an increase of the model's validity.

(14) **A System to Support Attention Allocation: Development and Application**

Research methodology: In Chapter 14 a description of a more elaborate version of the in Chapter 11 described attention allocation support system was given (g). This support system was based on four different models related to attention and the manipulation of human attention, which were first described (b) and then formalized (d). The first model described the human's current attentional state (as described in Chapter 10), the second was a model for beliefs about the human's attentional state, the third was a model to determine the discrepancy between the estimated current (descriptive) and normative (prescriptive) attentional state and the last was a model for the manipulation of the human's attention. Based on a simulation, several expected model properties of the above mentioned models (e) as well as the attention allocation support system were verified (h). Also, the support system was evaluated using performance data of humans executing a task (i). Like in Chapter 11, the task used was the tactical picture compilation task that was first introduced in Chapter 9.

Main conclusions: The participants reported to be confident that the agent's manipulation indeed was helpful. The results of the validation study with respect to performance improvement were positive. A detailed analysis and verification of the behavior of the agent also provided positive results: First, checking of the traces of the experiment confirmed that the agent was able to adapt the features of different objects in the task in such a way that they attracted human attention. The results furthermore showed that when there was a discrepancy between the prescriptive and the descriptive model of attention, the agent indeed was able to attract the human's attention.

Future research: Further investigation was needed to rule out possible order effects in the results of the described experiment, which suggests more research with more participants. It was also expected that future improvements of the agent's four sub-models, based on the gained knowledge from automated verification will also contribute to the improved success of such validation experiments. Top-down influences were not taken into account in the current models, but previous research shows that it is possible to extend such models based on saliency maps with top-down features of attention (see e.g., Elazari and Itty, 2010; Navalpakkam and Itti, 2002). As the presented attention model was based on the generic notion of features of a location, it could be easily extended with top-down features as well. In the future, these possibilities need to be explored in detail.

(15) **Adaptive Attention Allocation Support: Effects of System Conservativeness and Human Competence**

Research methodology: Finally, in Chapter 15 three variants (the fixed, lib-

eral and conservative support) of the in Chapter 14 developed support were described (g), based on which several psychological hypotheses (c) as well as hypotheses related to the effectiveness of the support (i) were formed and experimentally tested. As in Chapter 14, all support types assisted humans in their allocation of attention. The variants of support were different with respect to their conservativeness (i.e., tendency to support). In *fixed support*, the system calculated an estimated optimal decision and suggested this to the human. In the other two support types, the system estimated the important information in the problem space in order to make a correct decision and directed the human's attention to this information. In *liberal support*, the system attempted to direct the human's attention using only the assessed task requirements, whereas in *conservative support*, the this attempt was done provided that it was estimated that the human was not already paying attention (more conservative).

Main conclusions: Overall results did not confirm our hypothesis that adaptive conservative support leads to the best performances. Furthermore, especially high-competent humans showed more trust in a system when delivered support was adapted to their specific needs.

Future research: Working with complex (support) systems can raise the cognitive load on the human, leaving less capacity to focus on the actual monitoring of contacts. Future design of adaptive support systems should therefore aim at keeping the system as simple as possible, though preserving the expected advantages of adaptivity. For the adaptive support investigated in this study, it was not possible for the human to simply follow suggestions of the support system. This was because, instead of suggesting a possible answer to a problem, only areas of interest were indicated by the system. This meant that, in any case, the proposed adaptive support must have eliminated inappropriate reliance on the support. It was therefore believed that the found results in the study were not a reason for rejecting this principle and therefore more research on adaptive attention allocation support was suggested, focusing on the *requirements* in which such a system can help to gain task performance.

As can be concluded from the above descriptions of the relation between the different chapters and the methodological phases described in Chapter 1, indeed all phases of the proposed methodology have successfully been used at least once for both trust and attention. This would suggest that the used research methodology indeed was usable given the stated research objective at the beginning of this thesis.

The general discussion about the implications emerging from the in this section summarized main conclusions and future research is held in Section 2.

2 General Discussion

As the collection of the main conclusions summarized in the previous section might suggest: one research question can generate multiple answers. In this thesis several examples have been explored of adaptive human-computer team support based on cognitive models of trust and attention. For this reason, one could argue that indeed the objective stated in the beginning of this chapter has been reached. But as the collection of future research summarized in the previous section might also suggest: one answer can generate multiple research questions. And for that reason, one could also argue that there is still a very long way to go.

The cognitive models explored in this thesis focused on *trust* and *attention*. But as was mentioned in the introduction, there are many more cognitive functions, concepts or processes that would be very good candidates for the purpose of adapting automated support to the human state and capabilities. Future research might as well aim at the development and use of cognitive models that can closely predict situation awareness, vigilance, mode awareness, automation-induced complacency, mental load, boredom, emotion, skill, experience, stress, self-confidence and commitment (to name but a few), and determine their characteristics in terms of for example demand for transparency, system autonomy, task switching costs, responsibility, 'human in the loop'-ness, delegation strategy and organization characteristics. Further investigation might also imply alternatives for on-line parameter tuning (s.a. usage of profiles), eye-trackers and mouse devices (s.a. pupil size (for detecting timing of decisions), EEG (s.a. usage of the P300), skin response (arousal, lying detection) and ECG (workload)). The use of such objective measures as input for cognitive models is expected to be very useful, but one should keep in mind that these models easily result in low construct validity (i.e., the degree to which one is indeed estimating the actual psychological phenomenon). Furthermore, one could presume that the discrimination of different more detailed cognitive states are the way to go: these more detailed states can help fine-tune the adaptations to the human need for assistance. But there is, of course, a limit to the value of adding more detail to cognitive models, given the fact that eventually one is estimating the state of a black box, as our knowledge of the underpinnings of the human mind is still limited. Finally, the models used in this thesis are used for adaptive decision support, but they might very well be useful for other kinds of applications, such as for the simulation of human cognition for, for instance, testing new interfaces or displays in expensive machines, such as aircraft (usability testing). In this example, cognitive models can be a cheap alternative for using the 'think aloud protocol' on well-paid pilots in simulators, which is also much more intrusive and time consuming.

The general advantage of the usage of cognitive models, as compared to behavioral or environmental models, as a basis for adaptive support systems, is that the detection of potential performance degradation or dangers can be done in an earlier stage. Behavioral or environmental models can only detect errors after the first signs of the underlying mistakes are observable, because no inference is made of what possible cognitive states might be causing these mistakes. An example is the pilot who relies on his automatic pilot while the current weather conditions are very bad. A support system is more likely to prevent an accident from happening when it infers that the pilot in fact is over-relying

on his support system then when the first sign of a decreased altitude is detected. A disadvantage of cognitive models is that these models do not have a direct data source that can help in the inference of cognitive states (apart from for instance EEG, but still such sources are indirect). These sources do exist for behavioral and environmental models. For this reason, experimentally verified rules need to be identified that can substitute a direct data source for cognitive models. These rules are based on the fact that certain changes in the world can be antecedents for cognition and that cognition itself can be an antecedent of behavior. These two facts can be used to search for more specific behavioral and environmental data which help in the estimation of cognitive states and thereafter in the detection of limitations in human cognition.

A note on the scalability of this research. The reason for using different laboratory tasks was that the experiments can be controlled very well, participants could easily be measured (s.a. when using sensors like eye-trackers) and the experiment could be set up more easily, especially when multiple participants and computers were involved. But more realistic scenarios in which the results of these studies can scale up to real applications on, for instance, frigates or air traffic control towers, still need to be proven realizable. However, it is expected that the described studies and studies alike are a necessity when it comes to proper preparation for the further development of such systems.

A final note on the ethical implications of the research. It should be noted that the application of systems that are able to adapt to humans also need to *monitor* humans, *influence* their cognitive state and will *take over* tasks that formerly human beings were responsible for. These tasks can also be tasks that are about life and death. It is evident that such adaptive systems can have tremendous impact on society and, as a consequence, this should be subject for future ethical and political debates. Before technological advances can lead to the use of these adaptive support systems, both humans and systems should be ready for this: humans need to be ready on how to use and get used to such systems; and the systems need to be socially capable enough to take the human factor in human-computer cooperation into account, just like humans would do if they would stand in the shoes of the system (or even better than that).

Part V

Appendices

Bibliography

Journal publications

— Authors in alphabetical order —

Bosse, T., van Maanen, P.-P., and Treur, J. (2009f). Simulation and formal analysis of visual attention. *Web Intelligence and Agent Systems: An International Journal (WIAS)*, 7(1):89–105.

Hoogendoorn, M., Jonker, C., van Maanen, P.-P., and Sharpanskykh, A. (2008). Formal analysis of empirical traces in incident management. *Reliability Engineering and System Safety*, 97(10):1422–1433.

Hoogendoorn, M., Jonker, C., van Maanen, P.-P., and Treur, J. (2009). Agent-based analysis and simulation of meta-reasoning processes in strategic naval planning. *Knowledge-Based Systems (KnoSys)*, 22(8):589–599.

Proceedings publications

de Koning, L., van Maanen, P.-P., and van Dongen, K. (2008). Effects of task performance and task complexity on the validity of computational models of attention. In *Proceedings of the Human Factors and Ergonomics Society's 52nd Annual Meeting*.

Neef, M., van Maanen, P.-P., Petiet, P., and Spoelstra, M. (2009). Adaptive work-centered and human-aware support agents for augmented cognition in tactical environments. In *Proceedings of International Conference on Augmented Cognition, Jointly held with International Conference on Human-Computer Interaction*.

Nijholt, A., Meijerink, F., and van Maanen, P.-P. (2008). A virtual diary companion. In Wilks, Y., editor, *Proceedings of the Fourth International Workshop on Human-Computer Conversation*. University of Sheffield.

van Dongen, K. and van Maanen, P.-P. (2006a). Under-reliance on the decision aid: A difference in calibration and attribution between self and aid. In *Proceedings of the Human Factors and Ergonomics Society's 50th Annual Meeting*.

van Maanen, P.-P., de Koning, L., and van Dongen, K. (2008a). Design and validation of habta: Human attention-based task allocator. In Mühlhäuser, M., Ferscha, A., and Aitenbichler, E.,

editors, *Proceedings of the First International Workshop on Human Aspects in Ambient Intelligence*, volume 11 of *Communications in Computer and Information Science (CCIS)*, pages 286–300. Springer-Verlag.

van Maanen, P.-P., Klos, T., and van Dongen, K. (2007a). Aiding human reliance decision making using computational models of trust. In *Proceedings of the Workshop on Communication between Human and Artificial Agents (CHAA'07)*, pages 372–376, Fremont, California, USA. IEEE Computer Society Press. Co-located with The 2007 IEEE IAT/WIC/ACM International Conference on Intelligent Agent Technology.

van Maanen, P.-P., Klos, T., and van Dongen, K. (2007b). Closed-loop adaptive decision support based on automated trust assessment. In Schmorrow, D. D. and Reeves, L. M., editors, *Proceedings of the Third International Conference on Augmented Cognition (ACI) and 12th International Conference on Human-Computer Interaction (HCI'07)*, volume 4565 of *Lecture Notes in Computer Science*. Springer Verlag.

van Maanen, P.-P., Lindenberg, J., and Neerincx, M. A. (2005). Integrating human factors and artificial intelligence in the development of human-machine cooperation. In Arabnia, H. R. and Joshua, R., editors, *Proceedings of the 2005 International Conference on Artificial Intelligence (ICAI'05)*, volume 1, pages 10–16. CSREA Press.

van Maanen, P.-P. and van Dongen, K. (2005a). Towards task allocation decision support by means of cognitive modeling of trust. In Castelfranchi, C., Barber, S., Sabater, J., and Singh, M., editors, *Proceedings of the Eighth International Workshop on Trust in Agent Societies (Trust 2005)*, pages 168–77.

— Authors in alphabetical order —

Abbink, H., van Dijk, R., Dobos, T., Hoogendoorn, M., Jonker, C. M., Konur, S., van Maanen, P.-P., Popova, V., Sharpanskykh, A., van Tooren, P., Treur, J., Valk, J., Xu, L., and Yolum, P. (2004b). Automated support for adaptive incident management. In van de Walle, B. and Carle, B., editors, *Proceedings of the International Workshop on Information Systems for Crisis Response and Management '04*, pages 69–74.

Bosse, T., van Doesburg, W., van Maanen, P.-P., and Treur, J. (2007b). Augmented metacognition addressing dynamic allocation of tasks requiring visual attention. In Schmorrow, D. D. and Reeves, L. M., editors, *Proceedings of the Third International Conference on Augmented Cognition (ACI) and 12th International Conference on Human-Computer Interaction (HCI'07)*, volume 4565 of *Lecture Notes in Computer Science*. Springer Verlag.

Bosse, T., van Lambalgen, R., van Maanen, P.-P., and Treur, J. (2009b). Attention manipulation for naval tactical picture compilation. In *Proceedings of the 2009 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'09)*.

Bosse, T., van Lambalgen, R., van Maanen, P.-P., and Treur, J. (2009c). Automated visual attention manipulation. In Paletta, L. and Tsotsos, J., editors, *Proceedings of WAPCV'08, Attention in Cognitive Systems*, volume 5395 of *Lecture Notes in Computer Science*, pages 257–272. Springer-Verlag.

Bosse, T., van Maanen, P.-P., and Treur, J. (2006). A cognitive model for visual attention and its application. In Nishida, T., editor, *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT-06)*, pages 255–262. IEEE Computer Society Press.

Bosse, T., van Maanen, P.-P., and Treur, J. (2007d). Simulation and formal analysis of visual attention in cognitive systems. In *Proceedings of the Fourth International Workshop on Attention in Cognitive Systems (WAPCV'07)*. Published as: L. Paletta, E. Rome (Eds.), Attention in Cognitive Systems, Lecture Notes in AI, Springer Verlag, 2007.

Bosse, T., van Maanen, P.-P., and Treur, J. (2007e). Temporal differentiation of attentional processes. In Vosniadou, S. and Kayser, D., editors, *Proceedings of the Second European Cognitive Science Conference (EuroCogSci'07)*, pages 842–847. IEEE Computer Society Press.

Hoogendoorn, M., Jonker, C. M., Konur, S., van Maanen, P.-P., Popova, V., Sharpanskykh, A., Treur, J., Xu, L., and Yolum, P. (2004). Formal analysis of empirical traces in incident management. In Ellis, A. M. R. and Allen, T., editors, *Proceedings of the 24th International Conference on Innovative Techniques and Applications of Artificial Intelligence*, Lecture Notes in AI, pages 237–250. Stringer-Verlag.

Hoogendoorn, M., Jonker, C. M., van Maanen, P.-P., and Treur, J. (2006). An agent-based meta-level architecture for strategic reasoning in naval planning. In Kolp, M., Bresciani, P., Henderson-Sellers, B., and Winikoff, M., editors, *Agent-Oriented Information Systems III, Proceedings of the Seventh International Workshop on Agent-Oriented Information Systems (AOIS'05)*, volume 3529 of *Lecture Notes in AI*, pages 216–230. Springer Verlag.

van Lambalgen, R. and van Maanen, P.-P. (2010). Personalisation of computational models of attention by simulated annealing parameter tuning. In *Proceedings of the Fourth International Workshop on Human Aspects in Ambient Intelligence*. IEEE Computer Society Press. held in collaboration with the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology.

Extended abstracts

Greef, T. d. and van Maanen, P.-P. (2005). Automated adaptive support for task and information prioritizing. In Kaelbling, L. P. and Saffiotti, A., editors, *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI'05)*, pages 1713–4.

van Maanen, P.-P., de Koning, L., and van Dongen, K. (2008b). Design and validation of habta: Human attention-based task allocator (extended abstract). In *Proceedings of the 20th Belgium-Dutch Conference on Artificial Intelligence (BNAIC 2008)*.

van Maanen, P.-P. and van Dongen, K. (2005b). Towards task allocation decision support by means of cognitive modeling of trust (extended abstract). In Verbeeck, K., Tuyls, K., Nowe, A., Manderick, B., and Kuijpers, B., editors, *Proceedings of the 17th Belgian-Dutch Conference on Artificial Intelligence (BNAIC-2005)*, pages 399–400, Brussels, Belgium. Koninklijke Vlaamse Academie van Belgie voor Wetenschappen en Kunsten.

— Authors in alphabetical order —

Abbink, H., van Dijk, R., Dobos, T., Hoogendoorn, M., Jonker, C. M., Konur, S., van Maanen, P.-P., Popova, V., Sharpanskykh, A., van Tooren, P., Treur, J., Valk, J., Xu, L., and Yolum, P. (2004a). Automated support for adaptive incident management (extended abstract). In Verbrugge, R., Taatgen, and N., Schomaker, L., editors, *Proceedings of the 16th Belgian-Dutch Conference on Artificial Intelligence (BNAIC-2004)*, pages 349–350, Groningen, The Netherlands. University of Groningen.

Bosse, T., van Lambalgen, R., van Maanen, P.-P., and Treur, J. (2009d). Automated visual attention manipulation (extended abstract). In *Proceedings of the 21th Belgium-Dutch Conference on Artificial Intelligence (BNAIC 2009)*.

Bosse, T., van Lambalgen, R., van Maanen, P.-P., and Treur, J. (2009e). An interface agent for attention manipulation (extended abstract). In Decker, K., Sichman, J., Sierra, C., , and Castelfranchi, C., editors, *Proceedings of the Eighth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'09)*. ACM Press.

Bosse, T., van Maanen, P.-P., and Treur, J. (2007c). A cognitive model for visual attention and its application (extended abstract). In *Proceedings of the 19th Belgian-Dutch Conference on Artificial Intelligence (BNAIC 2007)*.

Hoogendoorn, M., Jonker, C. M., van Maanen, P.-P., and Treur, J. (2005a). An agent-based meta-level architecture for strategic reasoning in naval planning (extended abstract). In Verbeeck, K., Tuyls, K., Nowe, A., Manderick, B., and Kuijpers, B., editors, *Proceedings of the 17th Belgian-Dutch Conference on Artificial Intelligence (BNAIC-2005)*, pages 401–402, Brussels, Belgium. Koninklijke Vlaamse Academie van Belgie voor Wetenschappen en Kunsten.

Hoogendoorn, M., Jonker, C. M., van Maanen, P.-P., and Treur, J. (2005b). A meta-level architecture for strategic reasoning in naval planning (extended abstract). In Ali, M. and Esposito, F., editors, *Proceedings of the 18th International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems (IEA/AIE 2005)*, volume 3533 of *Lecture Notes in AI*, pages 848–850. Stringer Verlag.

Technical reports

van Maanen, P.-P., de Koning, L., and Meer, J. v. (2009). Factoren van invloed op ethische besluitvorming bij de nederlandse krijgsmacht. Technical Report TNO-DV-2009-B369, TNO Human Factors. English version submitted to journal.

van Maanen, P.-P., Keeris, E., Gaillard, A., Wetzer, I., and Six, C. (2008c). Using games to study support and training of decision making under stress. Technical Report TNO-DV-2008-IN266, TNO Human Factors.

Other publications

Meijerink, F., van Maanen, P.-P., Vliet, A. J. v., and Nijholt, A. (2007). Disclosure with an emotional intelligent synthetic partner. In *Workshop: Tools for Psychological Support during Exploration Missions to Mars and Moon*.

van Dongen, K. and van Maanen, P.-P. (2005). Designing for dynamic task allocation. In Schraagen, J. M. C., editor, *Proceedings of the Seventh International Naturalistic Decision Making Conference (NDM7)*.

van Dongen, K. and van Maanen, P.-P. (2006b). Under-trust in decision support systems. In *Proceedings of the Advice and Trust in Decision Making Conference (ATDM)*.

van Maanen, P.-P. and van Dongen, K. (2009). Realizing confidence in autonomous systems using automated trust assessment. In *Proceedings of IST-072 Trust and Confidence in Autonomous Systems, NATO Research and Technology Organisation*.

— Authors in alphabetical order —

Hoogendoorn, M. and van Maanen, P.-P. (2009). AI en HA. Human Ambience: De omgeving als beste vriend. *De Connectie*, 4(2):8–11.

Other References in Parts I and IV

Anderson, J. R. and Lebiere, C. (1998). *The atomic components of thought*. Erlbaum, Mahwah, NJ.

Bosse, T., Jonker, C., van der Meij, L., Sharpanskykh, A., and Treur, J. (2009a). Specification and verification of dynamics in agent models. *International Journal of Cooperative Information Systems*, 18:167–193.

Bosse, T., Jonker, C., van der Meij, L., and Treur, J. (2007a). LEADSTO: a Language and Environment for Analysis of Dynamics by SimulaTiOn. *International Journal of Artificial Intelligence Tools*, 16(3):435–464.

Dastani, M. (2008). 2APL: A practical agent programming language. *International Journal of Autonomous Agents and Multi-Agent Systems*, 16(3):214–248. Special Issue on Computational Logic-based Agents.

Dzindolet, M. T., Peterson, S. A., Pomransky, R. A., Pierce, L. G., and Beck, H. P. (2003). The role of trust in automation reliance. *International Journal of Human Computer Studies*, 58(6):697–718.

Elazari, L. and Itty, L. (2010). A bayesian model for efficient visual search and recognition. *Vision Research*, 50:1338–1352.

Itti, L. and Koch, C. (2001). Computational modeling of visual attention. *Nature Reviews Neuroscience*, 2(3):194–203.

Itti, L., Koch, U., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:1254–1259.

Kurzweil, R. (2005). *The Singularity is Near*. Pinguin Group, New York, NY.

Laird, J. E., Newell, A., and Rosenbloom, P. S. (1987). SOAR: an architecture for general intelligence. *Artificial Intelligence*, 33(1):1–64.

Navalpakkam, V. and Itti, L. (2002). A goal oriented attention guidance model. In *Biologically Motivated Computer Vision*, volume 2525 of *Lecture Notes Computer Science*, pages 453–461. Springer.

Neerincx, M. A. (2003). Cognitive task load design: model, methods and examples. In Hollnagel, E., editor, *Handbook of Cognitive Task Design*, chapter 13, page 283305. Lawrence Erlbaum Associates, Mahwah, NJ.

Parasuraman, R., Sheridan, T. B., and Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics*, 30:286–297.

Sun, R. (2002). *Duality of the Mind: A Bottom-up Approach Toward Cognition*. Lawrence Erlbaum Associates, Mahwah, NJ.

Sun, Y. (2003). *Hierarchical Object-Based Visual Attention for Machine Vision*. PhD thesis, University of Edinburgh.

Acknowledgments

It began back in January 2004 when I started my PhD research at the Vrije Universiteit Amsterdam (VUA) on the topic of Organization Dynamics. I just graduated from Utrecht University and it was dr. Frank Dignum and prof.dr. John-Jules Meyer who encouraged me to apply for a PhD student position. Prof.dr. Jan Treur and prof.dr. Catholijn Jonker at the VUA gave me such a position. If it were not for all of them, this dissertation would not exist at all.

The primary acknowledgments go to Jan Treur, for he was my supervisor (promotor). He has always been very consistent, concise and available. I could count on him for any advice in any situation. Catholijn has been my daily supervisor (copromotor) until she left the VUA and I would like to thank her for that period as well. She has been a very enthusiastic and knowledgeable coach for me. Something which was very important during my first months as a PhD student.

Later in 2004, Jan and I visited the Netherlands Organization for Applied Scientific Research (TNO) in Soesterberg to talk about a collaborative PhD research project. By coincidence we bumped into a very nice job vacancy for me at TNO (junior researcher), for which I applied almost immediately. Jan actually encouraged me, in spite of the fact that I had to leave the VUA for it. Michael Holewijn and prof.dr. Jan Maarten Schraagen offered me the job at TNO and I thank them for that. Thanks also to TNO in general for giving me the opportunity to develop myself further as researcher in different projects and write articles and reports that were often (not always!) in line with my PhD research topic. Without the very interesting, explorative and socially engaged research at TNO, it would never have been possible to write this thesis.

Since September 2007 I started as part-time researcher-lecturer at the VUA (for one day a week) and I would like to thank Jan again for giving me the opportunity to become a university teacher. The interaction with students is something I like very much and was something that I missed at TNO. It was also around that time that dr. Tibor Bosse came more into the picture as daily/weekly supervisor (copromotor). Tibor has become an upright friend and is great at decomposing complex tasks into smaller manageable subtasks. This helps a lot when you want to finish your thesis. Many thanks go to him as well! I should also acknowledge dr. Egon van den Broek for the period in which he was involved as daily/weekly supervisor: Thanks for being such a sharp and friendly person.

Of course, during the six and a half years at the VUA and six years at TNO it was also great to have worked with many other colleagues and friends with whom I share nice memories. I would say we have done quite some things together, such as: written papers,

organized workshops, were part of committees and reading groups, traveled abroad to conferences and attended summer schools and SIKS-courses. But we also had quite some nice parties, outings, spent holidays together, were great room mates, had serious and absurd discussions, drank beers in (sometimes not to be mentioned types of) bars, played volleyball, table tennis and in a band together. Many things that I will, of course, never forget and I am thankful to all of you for those nice moments!

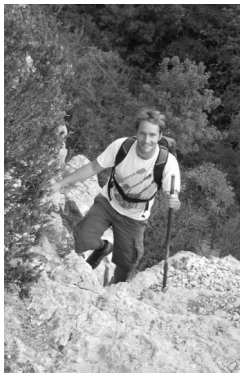
There are many people at work that were important while writing my dissertation and that I should mention specifically. At the VUA, special thanks go to (in alphabetical order) Alexei, Arlette, Andy, Annerieke, Azizi, Charlotte, Fiemke, Ghazanfar, Lai, Lourens, Mark van Assem, Mark Hoogendoorn, Martijn, Matthijs, Michael, Michel, Natalie, Nataliya, Pınar, Rianne, Rob, Robbert-Jan, Rogier, Savaş, Umair, Viara, Vera, Waqar and Zulfiqar. Thanks also to my friends from ProVU: The "promovendidagen" and "-borrels" were great! At TNO, special thanks go to Aletta, Anita, Anne-Marie, Carien, David, Eva, Frank, Guido, Ingrid, Jan van Erp, Jan-Willem Streefkerk, Jasper, John-Jules, Josephine, Josine, Joris, Jouke, Jurriaan, Karel, Kees van Dongen, Leo, Lex, Lisette, Maaike, Maarten, Maartje, Marc Grootjen, Mark Neerincx, Martijn Neef, Myra, Nanja, Peter Essens, Peter Petiet, Peter Rasker, Philippus, Rick, Rosemarijn, Tina, Tjerk, Tony Gaillard, Tony van Vliet, Willem and Wim. Thanks also to my friends at the Activities Committee of Jong TNO. Jong TNO is a great organization for having loads of fun with great people. Also thanks go to co-authors not from the VUA or TNO: Anton Nijholt and Tomas Klos. Many thanks to master's students Ferdi Meijerink, Florian Keiper, Francien Wisse, Sven Schomakers and Teun Lucassen: thanks for your cooperation and contribution to important parts of my research. Unfortunately it is impossible not to forget anybody in the above acknowledgments, so I apologize for those people I forgot. Thanks to you too!

Apart from the guys and girls at work, of course, things would have been much more worse if I did not have friends and family that made my life quite livable. Among them are Anne Willem, Arnout, Berend, Bram, Chee-Heun, Desiree, Jense, José, Koen, Rogier, Rutger, Roos, Selma, Theo and my brothers and sisters: Casper, Daniël, Esther and Hanneke. There was of course also a significant increase of the amount of joy in my life due to the existence of my brothers-, sisters- and parents-in-law: Geert and Kim, Marline and Marco (thanks for designing the cover of this book!), and Bert and Fransje. Everybody, please also stick around for the remaining part of my life!

I would also like to thank my parents, Magda and Peter, for their unconditional support and trust in me the past 30 years. Without you, nothing would have been possible (also the more important things in life than getting my PhD). Unfortunately my father did not live long enough to experience my life as it is now, but I am sure he would also be quite happy about that I finished writing this thesis. Also my grandmother has been a great support: her outright pride has always been heartwarming. My gratitude and love to all of you will never seize to exist.

Of course my final words go to my dearest Annelies. You support me in whatever insignificant thing I am worried about. This was yet another requirement for finalizing my dissertation. Thank you for caring about me. You have a Doctor as boyfriend now! ;-)

Curriculum Vitae



Peter-Paul van Maanen was born on the 31st of July 1980 in Raamsdonksveer. After graduating in 1998 at the 'Regionale Scholengemeenschap 't Rijks' in Bergen op Zoom, he went to study computer science (CS) at Utrecht University (UU). In 2003 he obtained his master's degree (with distinction) with specialization in artificial intelligence (AI) at UU. He wrote his thesis at the Institute of Cognitive Sciences and Technologies in Rome. Since September 2004, Peter-Paul is researcher and project manager at TNO Human Factors. In addition to his work at TNO, Peter-Paul is researcher-lecturer at the department of AI at the Vrije Universiteit Amsterdam (VUA) since September 2007. He is closely involved in several research projects at both TNO and the VUA. He co-authored publications and reports in research areas such as cognitive modeling, adaptive autonomy and support, trust and reliance decision making, emotion, serious games, attention, ethical decision making, unmanned vehicles and stress. In addition to his work, Peter-Paul likes to play the drums, to travel, to hike, to mountain bike, to squash, his friends and his girlfriend quite much.

Samenvatting

Adaptieve Ondersteuning van Mens-Computer Teams: Een Verkenning van het Gebruik van Cognitieve Modellen van Vertrouwen en Aandacht

In domeinen waarin veel geautomatiseerd is, zoals de luchtvaart, vliegverkeercentrales, kerncentrales en defensie, kan men een flink aantal uitdagingen vinden: Meer complexe missies, minder bemanning, hogere informatiedichtheid, hogere computerautonomie, meer ambiguïteit, meer tijdsdruk en hogere eisen voor wat betreft de samenwerking, zorgen voor een grote afstand tussen het geautomatiseerde (computers) en het niet-geautomatiseerde (mensen). Eén van de voornaamste problemen is dat zowel het niet-geautomatiseerde als het geautomatiseerde niet bewust is van de vermogens en onvermogens van de ander, terwijl ze sterk van elkaar afhankelijk zijn.

Samenwerkende mensen gedragen zich sociaal. Dat wil zeggen, ze schatten in wat de behoefte aan assistentie van de ander is en, afhankelijk van deze inschatting, passen daarna hun assistentie daarop aan. Hoewel in de state-of-the-art van human-computer interfaces steeds meer gebruikersmodellen worden ingezet om op een dergelijke manier proactief hulp aan te bieden, is er in de meeste gevallen nauwelijks sociaal gedrag in computerondersteuning te bekennen. Zeker interfaces die aansturen op het dynamisch en real-time aanpassen van assistentie aan de huidige toestand van de mens (tegenover het gebruik van voorgedefinieerde gebruikersprofielen) kunnen als relatief nieuwe technologie worden gezien.

In kritische situaties, kan het ontbreken van sociaal gedrag tussen mens en computer enorme gevolgen hebben. Een bekend voorbeeld hiervan is de vliegtuigpiloot die in zijn cockpit tegelijkertijd wordt geassisteerd door verschillende geautomatiseerde ondersteunende systemen. Deze overvloed aan ondersteuning leidt vaak tot een overvloed aan informatie, wat mogelijk kan leiden tot het niet zien van belangrijke informatie; bijvoorbeeld over het uitvallen van de motor van het vliegtuig. Een ander bekend voorbeeld is dat dezelfde vliegtuigpiloot te veel vertrouwt op de automatische piloot, terwijl het systeem geen rekening houdt met deze mogelijkheid van oververtrouwen, wat zelfs tot dodelijke gevolgen kan leiden. Het probleem dat naar voren komt bij deze twee voorbeelden is dat de piloten, of mensen in het algemeen, en hun ondersteunende systemen niet voldoende bewust zijn van de gevaren die het resultaat zijn van elkaars onvermogens. Het rekening

houden met deze onvermogens zou kunnen leiden tot een betere samenwerking tussen mens en computer.

Dit proefschrift gaat over het oplossen van het hierboven geschetste probleem (het ontbreken van sociaal gedrag tussen mens en machine), door het vermeerderen van het redeneervermogen van het ondersteunende systeem voor wat betreft hun eigen (on)vermogens en die van de mens(en) waarmee het samenwerkt (ook wel genoemd de 'menselijke factor'). Systeembewustzijn van, en adaptatie aan, deze (on)vermogens kan leiden tot een meer sociale en daarmee meer coöperatieve gedragingsvorm van het ondersteunende systeem. Systemen kunnen bijvoorbeeld bewust zijn van te grote informatiedichtheden, over- en ondervertrouwen, een te grote bevestigings- of automatiseringsbias, en cognitieve onder- en overbelasting. Op dit moment moeten mensen zelf specificeren in welke mate en hoe computers moeten assisteren (overigens is in veel gevallen zelfs dit al niet mogelijk). Echter, in de nabije toekomst zullen sociaal capabele ondersteunende systemen ook in staat zijn om deze specificatie uit te voeren voor zichzelf. Ze zullen zich automatisch aanpassen afhankelijk van de situatie, maar ook afhankelijk van de toestand van de menselijke gebruiker. Met name in tijdsgebonden situaties, zou dit de mens helpen in zijn moeilijke taak om systemen correct en tijdig te configureren, gegeven de huidige situatie. Dit zou tot betere prestaties moeten leiden doordat de mens dan meer tijd over heeft om zich op andere zaken te concentreren of doordat het systeem kennis neemt van (on)vermogens die anders onopgemerkt zouden blijven door de mens.

Het voorziene verhoogde redeneervermogen van de ondersteunende systemen wordt bereikt door in computertaal beschreven cognitieve modellen onderdeel te laten zijn van de systemen. Deze cognitieve modellen bevatten de benodigde kennis over de menselijke factor en schatten op elk moment in wat de huidige cognitieve toestand is van de menselijke gebruiker en of deze wenselijk is of niet. Op deze manier kan het systeem dus de kans op mogelijke fouten ontdekken voordat de fout zich überhaupt heeft gemanifesteerd. Deze inschattingen kunnen vervolgens gebruikt worden om een interventie te plegen om de mogelijke onwenselijke gevolgen te voorkomen. Zo'n interventie kan bestaan uit het geven van bepaalde informatie (ook wel *beslisondersteuning* genoemd) of het overnemen van delen van taken door de computer (ook wel *adaptieve autonomie* genoemd). Dit zou dan bijvoorbeeld kunnen leiden tot een verhoging van de mens-computer teamprestatie of een verbetering van iets anders wat men als doel heeft gesteld. De cognitieve modellen die gebruikt worden in dit proefschrift richten zich exclusief op 'vertrouwen' en 'aandacht'.

Vertrouwen is één van de belangrijkste regulatoren voor het gebruik van informatie tijdens het beslisproces van de mens. Het bepaalt onder andere of men wel of niet informatie of hulp van een ander (mens of computer) zal aannemen. Actuele kennis over vertrouwen kan worden ingezet om adaptief te bepalen hoe en wanneer men het beste ondersteuning kan geven om zo min mogelijk over- en ondervertrouwen te laten ontstaan. Nieuwe systemen zouden bijvoorbeeld adviseren niet op een automatische piloot te vertrouwen als wordt ingeschat dat men er te veel op vertrouwt en er slecht weer op komst is. Verder zou bijvoorbeeld ingeschat ondervertouwen kunnen leiden tot het duidelijker aangeven van een alarm als dit dringend genoeg is. Het systeem zou daarbij met behulp van meer argumentatie kunnen proberen de mens te overtuigen van de

urgentie van het voorval.

Aandacht is een cognitief proces dat belangrijk is bij de selectie en interpretatie van informatie van de 'externe wereld' (via de zintuigen) en 'interne wereld' (via gedachten). Dit betekent dat aandacht meer is dan alleen waar iemand naar kijkt: het heeft ook te maken met de vraag van welke objecten en onderwerpen iemand zich bewust is. Actuele kennis over aandacht kan gebruikt worden voor het zodanig aanpassen van ondersteuning dat deze past bij de objecten en onderwerpen waar de mens zich op dat moment op richt (of juist niet). Nieuwe systemen zouden bijvoorbeeld zelf kunnen bepalen wat de werkverdeling wordt tussen mens en computer, zonder dat de mens al te veel moeite hoeft te doen om de computer de juiste instructies te geven. Bijvoorbeeld bij het behandelen van vele contacten op een radarscherm, zou de computer de contacten over kunnen nemen die op dat moment niet worden behandeld door de mens.

De voornaamste uitkomsten van dit proefschrift zijn: 1) er is een generiek ontwerp van een adaptief ondersteunend systeem dat op bovengenoemde manier werkt voorgesteld en voor verschillende domeinen uitgewerkt, geimplementeerd en beproefd, en 2) er is een algemene methodologie voorgesteld waarmee men dergelijke adaptief ondersteunende systemen kan ontwikkelen op een manier die tot verbetering leidt.

SIKS Dissertation Series

1998

1998-1 Johan van den Akker (CWI)	DEGAS - An Active, Temporal Database of Autonomous Objects
1998-2 Floris Wiesman (UM)	Information Retrieval by Graphically Browsing Meta-Information
1998-3 Ans Steuten (TUD)	A Contribution to the Linguistic Analysis of Business Conversations within the Language/Action Perspective
1998-4 Dennis Breuker (UM)	Memory versus Search in Games
1998-5 E.W. Oskamp (RUL)	Computerondersteuning bij Straftoemeting

1999

1999-1 Mark Sloof (VU)	Physiology of Quality Change Modelling; Automated modelling of Quality Change of Agricultural Products
1999-2 Rob Potharst (EUR)	Classification using decision trees and neural nets
1999-3 Don Beal (UM)	The Nature of Minimax Search
1999-4 Jacques Penders (UM)	The practical Art of Moving Physical Objects
1999-5 Aldo de Moor (KUB)	Empowering Communities: A Method for the Legitimate User-Driven Specification of Network Information Systems
1999-6 Niek J.E. Wijngaards (VU)	Re-design of compositional systems
1999-7 David Spelt (UT)	Verification support for object database design
1999-8 Jacques H.J. Lenting (UM)	Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism for Discrete Reallocation

2000

2000-1 Frank Niessink (VU)	Perspectives on Improving Software Maintenance
2000-2 Koen Holtman (TUE)	Prototyping of CMS Storage Management
2000-3 Carolien M.T. Metselaar (UVA)	Sociaal-organisatorische gevolgen van kennistechnologie; een procesbenadering en actorperspectief
2000-4 Geert de Haan (VU)	ETAG, A Formal Model of Competence Knowledge for User Interface Design
2000-5 Ruud van der Pol (UM)	Knowledge-based Query Formulation in Information Retrieval
2000-6 Rogier van Eijk (UU)	Programming Languages for Agent Communication
2000-7 Niels Peek (UU)	Decision-theoretic Planning of Clinical Patient Management
2000-8 Veerle Coup (EUR)	Sensitivity Analysis of Decision-Theoretic Networks
2000-9 Florian Waas (CWI)	Principles of Probabilistic Query Optimization
2000-10 Niels Nes (CWI)	Image Database Management System Design Considerations, Algorithms and Architecture
2000-11 Jonas Karlsson (CWI)	Scalable Distributed Data Structures for Database Management

2001

2001-1 Silja Renooij (UU)	Qualitative Approaches to Quantifying Probabilistic Networks
2001-2 Koen Hindriks (UU)	Agent Programming Languages: Programming with Mental Models
2001-3 Maarten van Someren (UvA)	Learning as problem solving
2001-4 Evgueni Smirnov (UM)	Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets
2001-5 Jacco van Ossenbruggen (VU)	Processing Structured Hypermedia: A Matter of Style
2001-6 Martijn van Welie (VU)	Task-based User Interface Design
2001-7 Bastiaan Schonhage (VU)	Diva: Architectural Perspectives on Information Visualization
2001-8 Pascal van Eck (VU)	A Compositional Semantic Structure for Multi-Agent Systems Dynamics
2001-9 Pieter Jan 't Hoen (RUL)	Towards Distributed Development of Large Object-Oriented Models, Views of Packages as Classes
2001-10 Maarten Sierhuis (UvA)	Modeling and Simulating Work Practice BRAHMS: a multiagent modeling and simulation language for work practice analysis and design
2001-11 Tom M. van Engers (VUA)	Knowledge Management: The Role of Mental Models in Business Systems Design

2002

2002-01 Nico Lassing (VU)	Architecture-Level Modifiability Analysis
2002-02 Roelof van Zwol (UT)	Modelling and searching web-based document collections
2002-03 Henk Ernst Blok (UT)	Database Optimization Aspects for Information Retrieval
2002-04 Juan Roberto Castelo Valdueza (UU)	The Discrete Acyclic Digraph Markov Model in Data Mining
2002-05 Radu Serban (VU)	The Private Cyberspace Modeling Electronic Environments inhabited by Privacy-concerned Agents
2002-06 Laurens Mommers (UL)	Applied legal epistemology; Building a knowledge-based ontology of the legal domain
2002-07 Peter Boncz (CWI)	Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications
2002-08 Jaap Gordijn (VU)	Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas
2002-09 Willem-Jan van den Heuvel(KUB)	Integrating Modern Business Applications with Objectified Legacy Systems
2002-10 Brian Sheppard (UM)	Towards Perfect Play of Scrabble
2002-11 Wouter C.A. Wijngaards (VU)	Agent Based Modelling of Dynamics: Biological and Organisational Applications
2002-12 Albrecht Schmidt (UVA)	Processing XML in Database Systems
2002-13 Hongjing Wu (TUE)	A Reference Architecture for Adaptive Hypermedia Applications
2002-14 Wieke de Vries (UU)	Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems
2002-15 Rik Eshuis (UT)	Semantics and Verification of UML Activity Diagrams for Workflow Modelling
2002-16 Pieter van Langen (VU)	The Anatomy of Design: Foundations, Models and Applications
2002-17 Stefan Manegold (UVA)	Understanding, Modeling, and Improving Main-Memory Database Performance

2003

2003-01 Heiner Stuckenschmidt (VU)	Ontology-Based Information Sharing in Weakly Structured Environments
2003-02 Jan Broersen (VU)	Modal Action Logics for Reasoning About Reactive Systems
2003-03 Martijn Schuemie (TUD)	Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy
2003-04 Milan Petkovic (UT)	Content-Based Video Retrieval Supported by Database Technology

2003-05 Jos Lehmann (UVA) Causation in Artificial Intelligence and Law - A modelling approach
2003-06 Boris van Schooten (UT) Development and specification of virtual environments
2003-07 Machiel Jansen (UvA) Formal Explorations of Knowledge Intensive Tasks
2003-08 Yongping Ran (UM) Repair Based Scheduling
2003-09 Rens Kortmann (UM) The resolution of visually guided behaviour
2003-10 Andreas Lincke (UvT) Electronic Business Negotiation: Some experimental studies on the interaction between medium, innovation context and culture
2003-11 Simon Keizer (UT) Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks
2003-12 Roeland Ordelman (UT) Dutch speech recognition in multimedia information retrieval
2003-13 Jeroen Donkers (UM) Nosce Hostem - Searching with Opponent Models
2003-14 Stijn Hoppenbrouwers (KUN) Freezing Language: Conceptualisation Processes across ICT-Supported Organisations
2003-15 Mathijs de Weerdt (TUD) Plan Merging in Multi-Agent Systems
2003-16 Menzo Windhouwer (CWI) Feature Grammar Systems - Incremental Maintenance of Indexes to Digital Media Warehouses
2003-17 David Jansen (UT) Extensions of Statecharts with Probability, Time, and Stochastic Timing
2003-18 Levente Kocsis (UM) Learning Search Decisions

2004

2004-01 Virginia Dignum (UU) A Model for Organizational Interaction: Based on Agents, Founded in Logic
2004-02 Lai Xu (UvT) Monitoring Multi-party Contracts for E-business
2004-03 Perry Groot (VU) A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving
2004-04 Chris van Aart (UVA) Organizational Principles for Multi-Agent Architectures
2004-05 Viara Popova (EUR) Knowledge discovery and monotonicity
2004-06 Bart-Jan Hommes (TUD) The Evaluation of Business Process Modeling Techniques
2004-07 Elise Boltjes (UM) Voorbeeldig onderwijs; voorbeeldgestuurd onderwijs, een opstap naar abstract denken, vooral voor meisjes
2004-08 Joop Verbeek(UM) Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale politiële gegevensuitwisseling en digitale expertise
2004-09 Martin Caminada (VU) For the Sake of the Argument; explorations into argument-based reasoning
2004-10 Suzanne Kabel (UVA) Knowledge-rich indexing of learning-objects
2004-11 Michel Klein (VU) Change Management for Distributed Ontologies
2004-12 The Duy Bui (UT) Creating emotions and facial expressions for embodied agents
2004-13 Wojciech Jamroga (UT) Using Multiple Models of Reality: On Agents who Know how to Play
2004-14 Paul Harrenstein (UU) Logic in Conflict. Logical Explorations in Strategic Equilibrium
2004-15 Arno Knobbe (UU) Multi-Relational Data Mining
2004-16 Federico Divina (VU) Hybrid Genetic Relational Search for Inductive Learning
2004-17 Mark Winands (UM) Informed Search in Complex Games
2004-18 Vania Bessa Machado (UvA) Supporting the Construction of Qualitative Knowledge Models
2004-19 Thijs Westerveld (UT) Using generative probabilistic models for multimedia retrieval
2004-20 Madelon Evers (Nyenrode) Learning from Design: facilitating multidisciplinary design teams

2005

2005-01 Floor Verdenius (UVA) Methodological Aspects of Designing Induction-Based Applications
2005-02 Erik van der Werf (UM)) AI techniques for the game of Go
2005-03 Franc Grootjen (RUN) A Pragmatic Approach to the Conceptualisation of Language
2005-04 Nirvana Meratnia (UT) Towards Database Support for Moving Object data
2005-05 Gabriel Infante-Lopez (UVA) Two-Level Probabilistic Grammars for Natural Language Parsing
2005-06 Pieter Spronck (UM) Adaptive Game AI
2005-07 Flavius Frasincar (TUE) Hypermedia Presentation Generation for Semantic Web Information Systems
2005-08 Richard Vdovjak (TUE) A Model-driven Approach for Building Distributed Ontology-based Web Applications
2005-09 Jeen Broekstra (VU) Storage, Querying and Inferencing for Semantic Web Languages
2005-10 Anders Bouwer (UVA) Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments
2005-11 Elth Ogston (VU) Agent Based Matchmaking and Clustering - A Decentralized Approach to Search
2005-12 Csaba Boer (EUR) Distributed Simulation in Industry
2005-13 Fred Hamburg (UL) Een Computermodel voor het Ondersteunen van Euthanasiebeslissingen
2005-14 Borys Omelayenko (VU) Web-Service configuration on the Semantic Web; Exploring how semantics meets pragmatics
2005-15 Tibor Bosse (VU) Analysis of the Dynamics of Cognitive Processes
2005-16 Joris Graaumans (UU) Usability of XML Query Languages
2005-17 Boris Shishkov (TUD) Software Specification Based on Re-usable Business Components
2005-18 Danielle Sent (UU) Test-selection strategies for probabilistic networks
2005-19 Michel van Dartel (UM) Situated Representation
2005-20 Cristina Coteanu (UL) Cyber Consumer Law, State of the Art and Perspectives
2005-21 Wijnand Derks (UT) Improving Concurrency and Recovery in Database Systems by Exploiting Application Semantics

2006

2006-01 Samuil Angelov (TUE) Foundations of B2B Electronic Contracting
2006-02 Cristina Chisalita (VU) Contextual issues in the design and use of information technology in organizations
2006-03 Noor Christoph (UVA) The role of metacognitive skills in learning to solve problems
2006-04 Marta Sabou (VU) Building Web Service Ontologies
2006-05 Cees Pierik (UU) Validation Techniques for Object-Oriented Proof Outlines
2006-06 Ziv Baida (VU) Software-aided Service Bundling - Intelligent Methods & Tools for Graphical Service Modeling
2006-07 Marko Smiljanic (UT) XML schema matching – balancing efficiency and effectiveness by means of clustering
2006-08 Eelco Herder (UT) Forward, Back and Home Again - Analyzing User Behavior on the Web
2006-09 Mohamed Wahdan (UM) Automatic Formulation of the Auditor's Opinion
2006-10 Ronny Siebes (VU) Semantic Routing in Peer-to-Peer Systems
2006-11 Joeri van Ruth (UT) Flattening Queries over Nested Data Types
2006-12 Bert Bongers (VU) Interactivation - Towards an e-cology of people, our technological environment, and the arts
2006-13 Henk-Jan Lebbink (UU) Dialogue and Decision Games for Information Exchanging Agents
2006-14 Johan Hoorn (VU) Software Requirements: Update, Upgrade, Redesign - towards a Theory of Requirements Change
2006-15 Rainer Malik (UU) CONAN: Text Mining in the Biomedical Domain
2006-16 Carsten Riggelsen (UU) Approximation Methods for Efficient Learning of Bayesian Networks
2006-17 Stacey Nagata (UU) User Assistance for Multitasking with Interruptions on a Mobile Device
2006-18 Valentin Zhizhkun (UVA) Graph transformation for Natural Language Processing
2006-19 Birna van Riemsdijk (UU) Cognitive Agent Programming: A Semantic Approach
2006-20 Marina Velikova (UvT) Monotone models for prediction in data mining
2006-21 Bas van Gils (RUN) Aptness on the Web
2006-22 Paul de Vrieze (RUN) Fundaments of Adaptive Personalisation
2006-23 Ion Juvina (UU) Development of Cognitive Model for Navigating on the Web
2006-24 Laura Hollink (VU) Semantic Annotation for Retrieval of Visual Resources
2006-25 Madalina Drugan (UU) Conditional log-likelihood MDL and Evolutionary MCMC
2006-26 Vojkan Mihajlovic (UT) Score Region Algebra: A Flexible Framework for Structured Information Retrieval
2006-27 Stefano Bocconi (CWI) Vox Populi: generating video documentaries from semantically annotated media repositories
2006-28 Borkur Sigurbjornsson (UVA) Focused Information Access using XML Element Retrieval

2007

2007-01 Kees Leune (UvT)	Access Control and Service-Oriented Architectures
2007-02 Wouter Teepe (RUG)	Reconciling Information Exchange and Confidentiality: A Formal Approach
2007-03 Peter Mika (VU)	Social Networks and the Semantic Web
2007-04 Jurriaan van Diggelen (UU)	Achieving Semantic Interoperability in Multi-agent Systems: a dialogue-based approach
2007-05 Bart Schermer (UL)	Software Agents, Surveillance, and the Right to Privacy: a Legislative Framework for Agent-enabled Surveillance
2007-06 Gilad Mishne (UVA)	Applied Text Analytics for Blogs
2007-07 Natasa Jovanovic' (UT)	To Whom It May Concern - Addressee Identification in Face-to-Face Meetings
2007-08 Mark Hoogendoorn (VU)	Modeling of Change in Multi-Agent Organizations
2007-09 David Mobach (VU)	Agent-Based Mediated Service Negotiation
2007-10 Huib Aldewereld (UU)	Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols
2007-11 Natalia Stash (TUE)	Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System
2007-12 Marcel van Gerven (RUN)	Bayesian Networks for Clinical Decision Support: A Rational Approach to Dynamic Decision-Making under Uncertainty
2007-13 Rutger Rienks (UT)	Meetings in Smart Environments; Implications of Progressing Technology
2007-14 Niek Bergboer (UM)	Context-Based Image Analysis
2007-15 Joyca Lacroix (UM)	NIM: a Situated Computational Memory Model
2007-16 Davide Grossi (UU)	Designing Invisible Handcuffs. Formal investigations in Institutions and Organizations for Multi-agent Systems
2007-17 Theodore Charitos (UU)	Reasoning with Dynamic Networks in Practice
2007-18 Bart Orriens (UvT)	On the development an management of adaptive business collaborations
2007-19 David Levy (UM)	Intimate relationships with artificial partners
2007-20 Slinger Jansen (UU)	Customer Configuration Updating in a Software Supply Network
2007-21 Karianne Vermaas (UU)	Fast diffusion and broadening use: A research on residential adoption and usage of broadband internet in the Netherlands between 2001 and 2005
2007-22 Zlatko Zlatev (UT)	Goal-oriented design of value and process models from patterns
2007-23 Peter Barna (TUE)	Specification of Application Logic in Web Information Systems
2007-24 Georgina Ramrez Camps (CWI)	Structural Features in XML Retrieval
2007-25 Joost Schalken (VU)	Empirical Investigations in Software Process Improvement

2008

2008-01 Katalin Boer-Sorbn (EUR)	Agent-Based Simulation of Financial Markets: A modular,continuous-time approach
2008-02 Alexei Sharpanskykh (VU)	On Computer-Aided Methods for Modeling and Analysis of Organizations
2008-03 Vera Hollink (UVA)	Optimizing hierarchical menus: a usage-based approach
2008-04 Ander de Keijzer (UT)	Management of Uncertain Data - towards unattended integration
2008-05 Bela Mutschler (UT)	Modeling and simulating causal dependencies on process-aware information systems from a cost perspective
2008-06 Arjen Hommersom (RUN)	On the Application of Formal Methods to Clinical Guidelines, an Artificial Intelligence Perspective
2008-07 Peter van Rosmalen (OU)	Supporting the tutor in the design and support of adaptive e-learning
2008-08 Janneke Bolt (UU)	Bayesian Networks: Aspects of Approximate Inference
2008-09 Christof van Nimwegen (UU)	The paradox of the guided user: assistance can be counter-effective
2008-10 Wauter Bosma (UT)	Discourse oriented summarization
2008-11 Vera Kartseva (VU)	Designing Controls for Network Organizations: A Value-Based Approach
2008-12 Jozsef Farkas (RUN)	A Semiotically Oriented Cognitive Model of Knowledge Representation
2008-13 Caterina Carraciolo (UVA)	Topic Driven Access to Scientific Handbooks
2008-14 Arthur van Bunningen (UT)	Context-Aware Querying; Better Answers with Less Effort
2008-15 Martijn van Otterlo (UT)	The Logic of Adaptive Behavior: Knowledge Representation and Algorithms for the Markov Decision Process Framework in First-Order Domains
2008-16 Henriette van Vugt (VU)	Embodied agents from a user's perspective
2008-17 Martin Op 't Land (TUD)	Applying Architecture and Ontology to the Splitting and Allying of Enterprises
2008-18 Guido de Croon (UM)	Adaptive Active Vision
2008-19 Henning Rode (UT)	From Document to Entity Retrieval: Improving Precision and Performance of Focused Text Search
2008-20 Rex Arendsen (UVA)	Geen bericht, goed bericht. Een onderzoek naar de effecten van de introductie van elektronisch berichtenverkeer met de overheid op de administratieve lasten van bedrijven
2008-21 Krisztian Balog (UVA)	People Search in the Enterprise
2008-22 Henk Koning (UU)	Communication of IT-Architecture
2008-23 Stefan Visscher (UU)	Bayesian network models for the management of ventilator-associated pneumonia
2008-24 Zharko Aleksovski (VU)	Using background knowledge in ontology matching
2008-25 Geert Jonker (UU)	Efficient and Equitable Exchange in Air Traffic Management Plan Repair using Spender-signed Currency
2008-26 Marijn Huijbregts (UT)	Segmentation, Diarization and Speech Transcription: Surprise Data Unraveled
2008-27 Hubert Vogten (OU)	Design and Implementation Strategies for IMS Learning Design
2008-28 Ildiko Flesch (RUN)	On the Use of Independence Relations in Bayesian Networks
2008-29 Dennis Reidsma (UT)	Annotations and Subjective Machines - Of Annotators, Embodied Agents, Users, and Other Humans
2008-30 Wouter van Atteveldt (VU)	Semantic Network Analysis: Techniques for Extracting, Representing and Querying Media Content
2008-31 Loes Braun (UM)	Pro-Active Medical Information Retrieval
2008-32 Trung H. Bui (UT)	Toward Affective Dialogue Management using Partially Observable Markov Decision Processes
2008-33 Frank Terpstra (UVA)	Scientific Workflow Design; theoretical and practical issues
2008-34 Jeroen de Knijf (UU)	Studies in Frequent Tree Mining
2008-35 Ben Torben Nielsen (UvT)	Dendritic morphologies: function shapes structure

2009

2009-01 Rasa Jurgelenaite (RUN)	Symmetric Causal Independence Models
2009-02 Willem Robert van Hage (VU)	Evaluating Ontology-Alignment Techniques
2009-03 Hans Stol (UvT)	A Framework for Evidence-based Policy Making Using IT
2009-04 Josephine Nabukenya (RUN)	Improving the Quality of Organisational Policy Making using Collaboration Engineering
2009-05 Sietse Overbeek (RUN)	Bridging Supply and Demand for Knowledge Intensive Tasks - Based on Knowledge, Cognition, and Quality
2009-06 Muhammad Subianto (UU)	Understanding Classification
2009-07 Ronald Poppe (UT)	Discriminative Vision-Based Recovery and Recognition of Human Motion
2009-08 Volker Nannen (VU)	Evolutionary Agent-Based Policy Analysis in Dynamic Environments
2009-09 Benjamin Kanagwa (RUN)	Design, Discovery and Construction of Service-oriented Systems
2009-10 Jan Wielemaker (UVA)	Logic programming for knowledge-intensive interactive applications
2009-11 Alexander Boer (UVA)	Legal Theory, Sources of Law & the Semantic Web
2009-12 Peter Massuthe (TUE)	Operating Guidelines for Services
2009-13 Steven de Jong (UM)	Fairness in Multi-Agent Systems
2009-14 Maksym Korotkiy (VU)	From ontology-enabled services to service-enabled ontologies (making ontologies work in e-science with ONTO-SOA)
2009-15 Rinke Hoekstra (UVA)	Ontology Representation - Design Patterns and Ontologies that Make Sense
2009-16 Fritz Reul (UvT)	New Architectures in Computer Chess
2009-17 Laurens van der Maaten (UvT)	Feature Extraction from Visual Data
2009-18 Fabian Groffen (CWI)	Armada, An Evolving Database System
2009-19 Valentin Robu (CWI)	Modeling Preferences, Strategic Reasoning and Collaboration in Agent-Mediated Electronic Markets
2009-20 Bob van der Vecht (UU)	Adjustable Autonomy: Controling Influences on Decision Making

2009-21 Stijn Vanderlooy (UM) Ranking and Reliable Classification
2009-22 Pavel Serdyukov (UT) Search For Expertise: Going beyond direct evidence
2009-23 Peter Hofgesang (VU) Modelling Web Usage in a Changing Environment
2009-24 Annerieke Heuvelink (VU) Cognitive Models for Training Simulations
2009-25 Alex van Ballegooij (CWI) "RAM: Array Database Management through Relational Mapping"
2009-26 Fernando Koch (UU) An Agent-Based Model for the Development of Intelligent Mobile Services
2009-27 Christian Glahn (OU) Contextual Support of social Engagement and Reflection on the Web
2009-28 Sander Evers (UT) Sensor Data Management with Probabilistic Models
2009-29 Stanislav Pokraev (UT) Model-Driven Semantic Integration of Service-Oriented Applications
2009-30 Marcin Zukowski (CWI) Balancing vectorized query execution with bandwidth-optimized storage
2009-31 Sofiya Katrenko (UVA) A Closer Look at Learning Relations from Text
2009-32 Rik Farenhorst (VU) and Remco de Boer (VU) Architectural Knowledge Management: Supporting Architects and Auditors
2009-33 Khiet Truong (UT) How Does Real Affect Affect Affect Recognition In Speech?
2009-34 Inge van de Weerd (UU) Advancing in Software Product Management: An Incremental Method Engineering Approach
2009-35 Wouter Koelewijn (UL) Privacy en Politiegegevens; Over geautomatiseerde normatieve informatie-uitwisseling
2009-36 Marco Kalz (OUN) Placement Support for Learners in Learning Networks
2009-37 Hendrik Drachsler (OUN) Navigation Support for Learners in Informal Learning Networks
2009-38 Riina Vuorikari (OU) Tags and self-organisation: a metadata ecology for learning resources in a multilingual context
2009-39 Christian Stahl (TUE, Humboldt-Universitaet zu Berlin) Service Substitution – A Behavioral Approach Based on Petri Nets
2009-40 Stephan Raaijmakers (UvT) Multinomial Language Learning: Investigations into the Geometry of Language
2009-41 Igor Berezhnyy (UvT) Digital Analysis of Paintings
2009-42 Toine Bogers Recommender Systems for Social Bookmarking
2009-43 Virginia Nunes Leal Franqueira (UT) Finding Multi-step Attacks in Computer Networks using Heuristic Search and Mobile Ambients
2009-44 Roberto Santana Tapia (UT) Assessing Business-IT Alignment in Networked Organizations
2009-45 Jilles Vreeken (UU) Making Pattern Mining Useful
2009-46 Loredana Afanasiev (UvA) Querying XML: Benchmarks and Recursion

2010

2010-01 Matthijs van Leeuwen (UU) Patterns that Matter
2010-02 Ingo Wassink (UT) Work flows in Life Science
2010-03 Joost Geurts (CWI) A Document Engineering Model and Processing Framework for Multimedia documents
2010-04 Olga Kulyk (UT) Do You Know What I Know? Situational Awareness of Co-located Teams in Multidisplay Environments
2010-05 Claudia Hauff (UT) Predicting the Effectiveness of Queries and Retrieval Systems
2010-06 Sander Bakkes (UvT) Rapid Adaptation of Video Game AI
2010-07 Wim Fikkert (UT) Gesture interaction at a Distance
2010-08 Krzysztof Siewicz (UL) Towards an Improved Regulatory Framework of Free Software. Protecting user freedoms in a world of software communities and eGovernments
2010-09 Hugo Kielman (UL) A Politiele gegevensverwerking en Privacy, Naar een effectieve waarborging
2010-10 Rebecca Ong (UL) Mobile Communication and Protection of Children
2010-11 Adriaan Ter Mors (TUD) The world according to MARP: Multi-Agent Route Planning
2010-12 Susan van den Braak (UU) Sensemaking software for crime analysis
2010-13 Gianluigi Folino (RUN) High Performance Data Mining using Bio-inspired techniques
2010-14 Sander van Splunter (VU) Automated Web Service Reconfiguration
2010-15 Lianne Bodenstaff (UT) Managing Dependency Relations in Inter-Organizational Models
2010-16 Sicco Verwer (TUD) Efficient Identification of Timed Automata, theory and practice
2010-17 Spyros Kotoulas (VU) Scalable Discovery of Networked Resources: Algorithms, Infrastructure, Applications
2010-18 Charlotte Gerritsen (VU) Caught in the Act: Investigating Crime by Agent-Based Simulation
2010-19 Henriette Cramer (UvA) People's Responses to Autonomous and Adaptive Systems
2010-20 Ivo Swartjes (UT) Whose Story Is It Anyway? How Improv Informs Agency and Authorship of Emergent Narrative
2010-21 Harold van Heerde (UT) Privacy-aware data management by means of data degradation
2010-22 Michiel Hildebrand (CWI) End-user Support for Access to Heterogeneous Linked Data
2010-23 Bas Steunebrink (UU) The Logical Structure of Emotions
2010-24 Dmytro Tykhonov Designing Generic and Efficient Negotiation Strategies
2010-25 Zulfiqar Ali Memon (VU) Modelling Human-Awareness for Ambient Agents: A Human Mindreading Perspective
2010-26 Ying Zhang (CWI) XRPC: Efficient Distributed Query Processing on Heterogeneous XQuery Engines
2010-27 Marten Voulon (UL) Automatisch contracteren
2010-28 Arne Koopman (UU) Characteristic Relational Patterns
2010-29 Stratos Idreos(CWI) Database Cracking: Towards Auto-tuning Database Kernels
2010-30 Marieke van Erp (UvT) Accessing Natural History - Discoveries in data cleaning, structuring, and retrieval
2010-31 Victor de Boer (UVA) Ontology Enrichment from Heterogeneous Sources on the Web
2010-32 Marcel Hiel (UvT) An Adaptive Service Oriented Architecture: Automatically solving Interoperability Problems
2010-33 Robin Aly (UT) Modeling Representation Uncertainty in Concept-Based Multimedia Retrieval
2010-34 Teduh Dirgahayu (UT) Interaction Design in Service Compositions
2010-35 Dolf Trieschnigg (UT) Proof of Concept: Concept-based Biomedical Information Retrieval
2010-36 Jose Janssen (OU) Paving the Way for Lifelong Learning; Facilitating competence development through a learning path specification
2010-37 Niels Lohmann (TUE) Correctness of services and their composition
2010-38 Dirk Fahland (TUE) From Scenarios to components
2010-39 Ghazanfar Farooq Siddiqui (VU) Integrative modeling of emotions in virtual agents
2010-40 Mark van Assem (VU) Converting and Integrating Vocabularies for the Semantic Web
2010-41 Guillaume Chaslot (UM) Monte-Carlo Tree Search
2010-42 Sybren de Kinderen (VU) Needs-driven service bundling in a multi-supplier setting - the computational e3-service approach
2010-43 Peter van Kranenburg (UU) A Computational Approach to Content-Based Retrieval of Folk Song Melodies
2010-44 Pieter Bellekens (TUE) An Approach towards Context-sensitive and User-adapted Access to Heterogeneous Data Sources, Illustrated in the Television Domain
2010-45 Vasilios Andrikopoulos (UvT) A theory and model for the evolution of software services
2010-46 Vincent Pijpers (VU) e3alignment: Exploring Inter-Organizational Business-ICT Alignment
2010-47 Chen Li (UT) Mining Process Model Variants: Challenges, Techniques, Examples
2010-48 Milan Lovric Behavioral Finance and Agent-Based Artificial Markets
2010-49 Jahn-Takeshi Saito (UM) Solving difficult game positions
2010-50 Bouke Huurnink (UVA) Search in Audiovisual Broadcast Archives
2010-51 Alia Khairia Amin (CWI) Understanding and supporting information seeking tasks in multiple sources
2010-52 Peter-Paul van Maanen (VU) Adaptive Support for Human-Computer Teams: Exploring the Use of Cognitive Models of Trust and Attention